Lokianalyysi ja valvonta ELK-järjestelmää hyödyntäen

Jouni Mikkola

2020 Laurea

Laurea-ammattikorkeakoulu

Lokianalyysi ja valvonta ELK-järjestelmää hyödyntäen

Jouni Mikkola
Tietojenkäsittely
Opinnäytetyö
Tammikuu, 2020 2020

Laurea-ammattikorkeakoulu **Tiivistelmä**
Tietojenkäsittely
Tradenomi (AMK)

Jouni Mikkola

Lokianalyysi ja valvonta ELK-järjestelmää hyödyntäen

2020 2020 Sivumäärä 58

Lokidatan analysointiin käytetään usein tehottomia menetelmiä ja järjestelmiä, jotka saattavat helpottaa datan suodatusta, mutta ei välttämättä tehosta analysointia muuten. Tämän työn tarkoituksena on tutkia, että pystytäänkö lokianalyysiä ja lokivalvontaa tehostaa käyttäen ELK-järjestelmää.

Kehittämistehtävänä oli parantaa kirjoittajan omaa tietoisuutta ELK-järjestelmän ominaisuuksista lokianalyysin ja valvonnan suorittamiseen. Lisäksi valmis opinnäytetyö tulee julkiseen jakeluun ja täten työtä voidaan käyttää kaikkien aiheesta kiinnostuneiden tietämyksen parantamiseen. Työn tarkoitus ei ole olla käyttöopas ELK-järjestelmän käyttöönottoon, vaan toteuttaa lokilähteiden integrointeja sillä tasolla, että tuotteen ominaisuuksia voidaan analysoida.

Tietoperustana työssä toimi suurimmalta osin sähköinen materiaali, koska työn luonteen huomioiden painettu materiaali ei yleensä ole yhtä ajantasaista kuin sähköisesti saatavilla oleva tieto. Työssä kuitenkin hyödynnettiin myös painettua materiaalia, silloin kun se oli järkevää.

Opinnäytetyö on luonteeltaan kehitystyö, jossa kehitetään kotiverkon valvontaa ELK-järjestelmällä. Kehittämismenetelmänä käytettiin osallistuvaa havainnointia. Menetelmä soveltuu tähän opinnäytetyöhön hyvin, koska tutkija asentaa ELK-järjestelmän itse, jolloin tutkija osallistuu, sekä analysoi tuloksia. Kehitystyössä ELK-järjestelmä asennettiin valvomaan kotiverkkoa hyödyntäen lähdekirjallisuutta. Tuotteen kyvykkyyttä valvoa kotiverkkoa analysoitiin hyödyntäen SWOT-analyysiä.

Työssä kartoitettiin useita ELK:n pohjautuvia järjestelmiä, joiden perusteella valittiin järjestelmä, jonka asennus toteutettiin. Toteutettu ELK-järjestelmä asennettiin valvomaan suhteellisen normaalia kotiverkkoympäristöä, joskin ympäristössä on hivenen enemmän laitteita kuin perinteisessä kotiverkossa on.

Lopputuloksena työssä havaittiin, että ELK-järjestelmä pystyy tehostamaan lokianalyysiä, sekä lokivalvontaa huomattavan paljon. Työ osoitti, että järjestelmän keskeiset toiminnallisuudet sopivat lokidatan analysoimiseen erittäin hyvin ja se tarjoaa ominaisuuksia, kuten datan visualisointi, mikä tekee analysoimisesta huomattavasti helpompaa. Lisäksi työ osoitti, että käyttöönotto ei ole välttämättä hirvittävän hankalaa, mutta lokilähteiden integroinnit voivat olla hyvin erilaisia ja osittain erittäin työläitä toteuttaa.

Jatkokehityksenä olisi hyvä tehdä lisätutkimuksia siitä, että miten hyvin tuotteeseen voidaan integroida pilvipohjaisia lokilähteitä, sekä miten tuotteessa voitaisiin ottaa käyttöön korreloiva koneoppimiseen perustuva ominaisuus, joka on saatavilla vain maksullisen lisenssin takana.

Asiasanat: Loki, lokianalyysi, lokivalvonta, ELK

Laurea University of Applied Sciences **Abstract**
Degree Programme in Business Information Technology
Bachelor's Degree

Jouni Mikkola

Log analysis and monitoring using ELK

2020 2020 Pages 58

Log analysis and monitoring are often done by inefficient methods and tools, which offers limited functionality. This thesis aims to research if this can be made more efficient with Elasticsearch, Logstash, Kibana Stack (ELK).

The development task was to improve the author's own knowledge of the ELK system and to provide an community with insight to the topic as when the thesis is published data was retrieved mainly from electronic sources which are often more up to date, but printed data was used when possible. The development method used in this thesis was participatory observation i.e. the author built the security monitoring system from ground up while analysing the outcomes. The ELK system was installed with the help of using source literature. The capabilities of the system were analysed with help of a SWOT analysis.

The thesis included a comparison of a few ELK-based systems from which the chosen ELK system was installed to monitor the home network of the author. The conclusion is that ELK can make log analysis and monitoring much more efficient with the provided visualization tools but further study is required to analyse if cloud environments can be monitored and how the ELK system could be used to correlate data and create alerts.

Keywords: Logs, log analysis, log monitoring, ELK

Sisällys

1 Johdanto 7
2 Opinnäytetyön lähtökohdat 7
2.1 Kehittämistavoitteet 8
2.2 Keskeiset käsitteet 8
2.3 Opinnäytetyön rajaukset 9
3 Lokianalyysijärjestelmät 10
3.1 Lokianalyysi 10
3.2 Lokivalvonta 12
4 Tutkimus- ja kehittämismenetelmät 13
4.1 Kehittämistutkimus 13
4.2 Käytännön toteutus ja testaus 14
4.3 Lähdekirjallisuus 14
4.4 SWOT-analyysi 15
4.5 Sisällönanalyysi 15
4.6 Validiteetti ja reliabiliteetti 16
5 ELK-järjestelmän käyttöönotto 16
5.1 ELK -järjestelmän kuvaus 16
5.1.1 Beats 17
5.1.2 Logstash 18
5.1.3 Elasticsearch 18
5.1.4 Kibana 19
5.2 Asennettavan ELK järjestelmän valinta 19
5.2.1 ELK:in vahvuudet ja heikkoudet 20
5.2.2 SOF-ELK 20
5.2.3 Security Onion 23
5.2.4 Logz.io 24
5.2.5 Yhteenveto ja valinnan perustelu 25
5.3 Asennettavan ympäristön kuvaus ja käyttötarkoitus 26
5.4 Asennussuunnitelma ja asennus 28
5.5 Järjestelmän asetukset 30
5.5.1 Logstash asetukset 30
5.5.2 Elasticsearch asetukset 30
5.5.3 Kibana asetukset 32
5.5.4 Filebeat asetukset 32
5.6 Lokilähteiden läpikäynti ja integrointi järjestelmään 33
5.6.1 Reitittimen syslog 34

5.6.2 Verkkohälytystiedot 35
5.6.3 Verkkovalvonta ja analysointi 35
5.6.4 Päätelaitevalvonta 36
5.6.5 Palvelinten pääsynhallinnan lokit 36
5.6.6 Palvelinten suorituskykytiedot 37
5.7 Tallennettujen hakujen luonti lokilähteiden tarpeisiin 38
5.8 Visualisointien luonti 40
5.9 Yhtenäisen käyttöliittymän luominen (”Dashboard”) 42
6 Asennetun ELK -järjestelmän kuvaus ja yhteenveto 48
7 SWOT-analyysi 49
7.1 SWOT-analyysin vahvuudet 50
7.2 Mahdollisuudet 50
7.3 Uhat 51
7.4 Heikkoudet 51
8 Johtopäätökset 51
9 Jatkokehitysehdotukset 53
Lähteet 54
Kuviot 57
Taulukot 58

1 Johdanto

Jatkuvasti lisääntyvä järjestelmien määrä luo suurempaa tarvetta tehdä erinäköisiä analyyseja, jotka liittyvät näiden järjestelmien toimivuuteen sekä tietoturvaan. Tätä varten lähes kaikki järjestelmät tuottavat toiminnastaan ja tapahtumistaan erilaista lokitietoa.

Lokit ovat nykypäivänä avainasemassa monissa eri (tietotekniikan) analyyseissa ja tutkimuksissa, mutta lokien analyysia suoritetaan usein melko tehottomilla välineillä, sekä metodeilla, joita voitaisiin tehostaa käyttämällä avuksi järjestelmää, joka pystyy käsittelemään suurta määrää dataa nopeasti. Lokianalyysilla voidaan tuottaa paljon erilaista arvoa organisaatioille, kuten esimerkiksi paljastaa sovelluksissa olevan suorituskykyongelman.

Lokit ovat erilaisia tiedostoja, joita sovellukset kirjoittavat. Lokit sisältävät dataa liittyen sovellukseen, joka kirjoittaa lokitietoa lokitiedostoon. Monesti, järjestelmät tuottavat vähintäänkin lokia virheistä, eli jos sovelluksessa tapahtuu jokin virhe, niin tästä kirjoitetaan yksi rivi lokitiedoston jatkeeksi. Monesti sovellukset kirjoittavat myös pääsynhallinnan lokia, eli lokia siitä, kun jonkin tunnuksen kirjautuminen sovellukseen sallitaan – tai ei sallita. Tämän tyyppiset lokit ovatkin erittäin tärkeitä tietoturvan valvonnassa, sekä tietoturva-analyyseissa sovellusten käytössä.

Tämän työn tarkoitus ei ole käsitellä lokitiedostoja itsessään kovin syvällisesti, vaikka aihealuetta käydään hieman läpi pohjustuksena. Opinnäytetyössä tehdään kotiverkon tietoturvavalvonnan kehitystä asentamalla kotiverkkoon ELK-järjestelmä, joka kerää kotiverkon laitteista lokeja ja niitä voidaan jatkoanalysoida järjestelmällä. ELK lyhenne tulee sanoista Elasticsearch, Logstash ja Kibana. Nämä ovat kolme erillistä komponenttia, jotka muodostavat ELK nimisen kokonaisuuden. Useat eri järjestelmät käyttävät taustalla ELK-järjestelmää, joten työssä analysoidaan useita tarkoitukseen sopivia järjestelmiä, joista valitaan soveltuvin opinnäyteyön luonteen kannalta.

Työn tarkoitus on selvittää, miten lokianalyysia voitaisiin tehostaa ja miten ELK -järjestelmä voisi auttaa sekä lokianalyysissä, että lokivalvonnassa. Työssä käsitellään lokianalyysia ja valvontaa lähinnä tietoturvanäkökulmasta, keskittyen lähteisiin, joita voitaisiin valvoa myös reaalimaailmassa.

2 Opinnäytetyön lähtökohdat

Opinnäytetyötä ei toteutettu millekään yritykselle, vaan sen tarkoitus on tukea tekijän omaa oppimista. Lisäksi opinnäytetyössä toteutettiin tekijän oman kotiverkon ja työasemaympäristön valvontajärjestelmän, jota voidaan hyödyntää myös yritystoteutuksessa. Opinnäytetyössä käytettävät sovellukset ovat avoimen lähdekoodin sovelluksia, joten ne ovat kenen tahansa vapaasti hyödynnettävissä omassa valvonnassaan.

Opinnäytetyön tarkoitus on selvittää, miten lokianalyysia voitaisiin tehostaa ja miten ELK -järjestelmä voisi auttaa sekä lokianalyysissä, että lokivalvonnassa. Opinnäytetyössä käsiteltiin lokianalyysia ja valvontaa lähinnä tietoturvanäkökulmasta, keskittyen lähteisiin, joita voitaisiin valvoa myös yritysympäristössä. Opinnäytetyöhön tarvittavat data tuotetaan kuluttaja-käyttäjän omista lähteistä, kuten kotiverkosta.

2.1 Kehittämistavoitteet

Opinnäytetyön tavoitteena on parantaa yritysten, yhteisöjen ja aiheesta kiinnostuneiden yksilöiden tuntemusta lokihallinnan järjestelmistä. Opinnäytetyöhön valittiin tuote jo ennen varsinaista analysointia, johtuen siitä, että ELK-sovelluksesta ei joudu maksamaan lisenssikustannuksia. Opinnäytetyössä esitellään erilaisia sovelluksia, jotka hyödyntävät jo valittua ELK-järjestelmää taustalla. Tällaisia järjestelmiä, hyödyntävät valittua ELK-järjestelmää taustalla on useita ja näistä merkittävämmät käydään läpi ja näistä valitaan paras työkalu suorittamaan analyysia ja valvontaa.

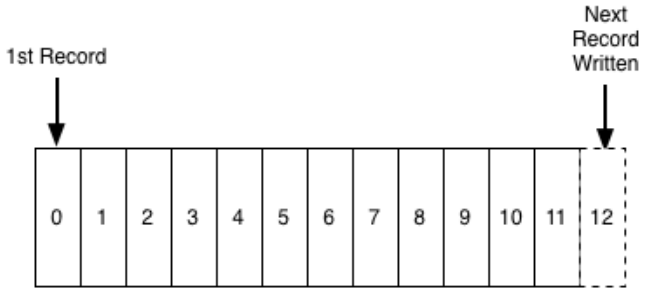
Toinen tavoite oli kehittää tekijän oman kotiverkon valvontaa, niin, että verkossa mahdollisesti tapahtuvat tietoturvapoikkeamat voidaan havaita. Tällä hetkellä kotiverkossa ei tehdä mitään valvontaa, joten työssä kehitetään uusi järjestelmä valvontaan ja ei korvata vanha.

Kolmas tavoite oli kehittää myös omaa tietoisuutta ja osaamista ELK:lla suoritettavasta lokianalyysista ja valvonnasta, sekä arvioida kuinka paljon lisäarvoa lokianalyysiin ja valvontaan voidaan tuottaa käyttäen hyväkseen siihen pohjautuvaa järjestelmää.

2.2 Keskeiset käsitteet

Opinnäytetyön kannalta oleellista on ymmärtää mitä ovat lokit. Loki itsessään ei välttämättä ole opinnäytetyön keskeinen käsite, mutta loki käsitteen ymmärtäminen on erittäin tärkeää, jotta opinnäytetyössä käsiteltävät keskeiset käsitteet voi ymmärtää.

Lokitapahtuma tarkoittaa käytännössä yksittäistä tapahtumaa, joka kertoo sovelluksen toiminnasta tai esimerkiksi pääsynhallinnasta. Lokitapahtumat kirjoitetaan aina järjestyksessä tapahtuman mukaan, eli ensimmäinen tapahtuma on kohdassa yksi, toinen kohdassa kaksi ja niin edelleen. Lokitapahtumissa tulisi olla myös aina aikaleimat, joilla lokitapahtumat voidaan järjestää. Lokitapahtumien kokonaisuus muodostaa varsinaisen lokin, joka siis käsittää useita lokitapahtumia. Lokien tarkoitus on siis kuvastaa mitä tapahtui ja milloin. (Kreps 2013.) Kuvio 1 kuvastaa miten lokitapahtumia kirjataan järjestyksessä lokiin.



Kuvio 1: Lokitapahtuman kirjoitus lokiin järjestyksessä (Kreps 2013.)

Lokeja tuotetaan monesta eri tapahtumasta ja näkökulmasta. Lokia voidaan kirjoittaa esimerkiksi web -palvelimen tapahtumista ja tämä onkin ehkä yksinkertaisin tapa hahmottaa miten lokit toimivat. Käytännössä, web-palvelin kirjoittaa jokaisen kutsun, joka tulee palvelimelle erilliseen tiedostoon järjestyksessä, sisältäen informaatiota tapahtumasta.

Informaation määrä ei ole vakio, vaan se vaihtelee riippuen siitä mikä web-palvelin on kyseessä. Microsoftin käyttämä IIS-web palvelin tuottaa erilaista lokia kuin vaikkapa Apache web -palvelin. Tämä on vain yksi esimerkki monien joukossa ja tämä opinnäytetyö kohdentuukin tietoturvamielessä merkitseviin lokeihin, kuten esimerkiksi pääsynhallinnan lokiin ja verkkotason lokiin.

Opinnäytetyössä keskeisenä käsitteenä on lokianalyysi, sekä lokivalvonta. Lokianalyysillä tarkoitetaan lokien sisältämien tapahtumien analysointia (Zhang 2018). Lokivalvonnalla tarkoitetaan lokien sisältämien tapahtumien automaattista valvontaa ilman manuaalista analyysiä (Logdna).

2.3 Opinnäytetyön rajaukset

Opinnäytetyöstä rajattiin pois erilaisten lokianalyysijärjestelmien vertailu. Näitä järjestelmiä on olemassa useita, joista tunnetuimpia on Splunk, ELK ja Graylog. Näiden kokonaisvaltainen vertailu olisi vienyt opinnäytetyön toteutuksen kannalta liikaa aikaa, sekä osa tuotteista vaatii lisenssimaksun, joten ne eivät olleet vaihtoehto. Tuotteiden vertailu olisi myös ollut omalta osaltaan hieman turhaa, koska osa tuotteista olisi rajautunut joka tapauksessa pois lisenssikustannuksen takia.

Opinnäytetyöstä rajattiin pois myös varsinaisten hälytyssääntöjen luonti, eli lokilähteitä valvottiin ELK-järjestelmällä, mutta mistään tapahtumista ei nostettu hälytyksiä. Hälytyssääntöjen tehokas luominen vaatii ELK-järjestelmälle tukisopimuksen, jonka hankkiminen opinnäytetyön toteuttamista varten ei ollut järkevää. Tämä vaatii myös huomattavan paljon erikoisosaamista, mikä tarkoittaa, että opinnäytetyön työmäärä olisi kasvanut liikaa.

Lisäksi rajattiin pois pilvipalveluiden valvonta, jota ensin suunniteltiin toteutettavaksi. Pilvipalvelut jätettiin pois, koska niiden integroiminen olisi ollut äärimmäisen vaikeaa suoran integrointikeinon puuttuessa. Valmis ratkaisu olisi ollut saatavilla, mutta se olisi vaatinut rahallista panostusta.

3 Lokianalyysijärjestelmät

Lokianalyysia ja valvontaa voi usein tehdä samalla järjestelmällä. Lokianalyysijärjestelmiä löytyy nykyään huomattavan suuri määrä, joten tässä käsitellään muutamia yleisimpiä, lyhyesti. Tämän opinnäytetyön osalta tuote valittiin jo ennen varsinaista vertailua, johtuen siitä, että valittu tuote on yksi yleisimmin käytetyistä, lisäksi se on ilmainen toisin kuin monet muut järjestelmät.

Lokianalyysijärjestelmiä on todella paljon nykyään, vaikka mietittäisiin vain suhteellisen nykyaikaisia järjestelmiä. Järjestelmien suuren määrän takia opinnäytetyön pohjustusta varten on valittu kolme tuotetta, jotka ovat hyvin tunnettuja. Tuotteet, joita opinnäytetyössä käydään läpi ovat Splunk, ELK ja Graylog. Nämä kaikki ovat erittäin tunnettuja ja suuressa käytössä eri organisaatioissa. Järjestelmissä on paljon samaa ja toimintalogiikka on hyvin samanlainen. Mikä tahansa näistä järjestelmistä olisi toiminut oikein hyvin.

Miksi siis ELK on valittu opinnäytetyöhön jo etukäteen, jos saatavilla on muitakin hyviä järjestelmiä kuten Splunk ja Graylog. Syynä tähän on se, että ELK on Splunkin tavoin erittäin tunnettu ja paljon käytetty järjestelmä. ELK on myös ilmainen ja se tukee JSON formaattia, joka helpottaa työskentelyä muiden järjestelmien integroinnin kanssa. Splunkin ilmainen versio on todella rajattu, Graylogin hieman vähemmän, mutta silti niin rajattu, että rajoitukset saattaisivat tulla vastaan opinnäytetyön puitteissa. Lisäksi Graylogin visualisointiominaisuudet eivät ole yhtä hyvät kuin Splunkissa tai Elkissä. (Bhargava 2018.)

3.1 Lokianalyysi

Lokianalyysi on lokien sisältämien tapahtumien evaluointia (Zhang 2018). Lokilähteet tulee usein myös tallentaa keskitetysti yhteen paikkaan, esimerkiksi Windows ja Unix palvelimilta, jotta lokianalyysia voidaan suorittaa keskitetysti (GadAllag 2004, 6).

Lokianalyysiä tehdään usein silloin, kun valvotaan laitteiden resurssien käyttöä tilanteessa, jossa laite tai sovellus toimii huonosti. Lokianalyysillä voidaan kohdentaa resurssi, joka on loppumassa, esimerkiksi palvelimen muisti. Lokianalyysilla voidaan myös tehdä ongelmanselvitystä. Web-palvelinten lokeissa esimerkiksi on usein virhekoodeja, jotka kertovat missä sovelluksessa on ongelmia. Tällöin lokianalyysilla voidaan selvittää, että mistä ongelmat johtuvat. (Kroupp.)

Kyberturva-alalla lokianalyysia tehdään erittäin paljon. Lokianalyysilla lokeista voidaan havaita kyberhyökkäyksiä tai uhkia. Yleensä kyberturva-alalla lokilähteitä on paljon ja niitä viedään erillisiin järjestelmiin. (Sumo Logic.) Tämä kuitenkin menee hieman enemmän lokivalvonnan puolelle, joka on toinen keskeinen käsite tässä opinnäytetyössä. Tätä käsitellään hieman myöhemmin.

Zhangin (2018) mukaan lokianalyysillä voidaan saavuttaa useita asioita, mutta tietoturvamielessä, lokianalyysiä käytetään usein seuraavissa käyttötapauksissa:

- Normaalin käyttötavan ymmärtäminen ja tämän takia anomalioiden tunnistus normaalin datan joukosta.
- Datan normalisointi, eli eri lokilähteiden datan yhtenäistäminen. Esimerkiksi, kellonajat ja päiväykset vaihtelevat ja ne tulee normalisoida eri lokilähteistä, että saadaan ajat täsmäämään.
- Datan kategoriointi käyttäen erilaisia kategorioita.
- Korrelaatio, eli eri lokilähteiden lukeminen niin, että ne liittyvät toisiinsa. Lokit voivat täydentää toisiaan ja korreloinnilla voidaan saada samaan tapahtumaan liittyvää dataa eri lähteistä. (Zhang, 2018.)

Tietoturvapoikkeamia on hyvin erilaisia, mutta usein niihin liittyy suuri määrä erilaisia lokeja. Lokimerkintöjä, joita tietoturvapoikkeamissa analysoidaan voivat olla esimerkiksi palvelimilta tai verkkolaitteilta kerättyä dataa ja se voi käsittää myös niin sanottua Net Flow dataa, joka on lokia siitä mitä verkossa on tapahtunut. Tämän tyyppistä dataa voidaan analysoida myös samalla järjestelmällä, jolla analysoidaan muutakin lokia. Lokianalyysillä pyritään etsimään tarkoin määrättyjä rivejä lokimassasta. Tällöin pyritäänkin suodattamaan dataa niin, että näkyville jää vain halutut rivit. (Balaji, 2019.)

Lokeja voi olla erittäin suuri määrä, usein voidaan puhua jopa useista sadoista gigatavuista. Tällöin lokeista pyritään selvittämään lisätietoa liittyen epäiltyyn tietoturvapoikkeamaan. Lokianalyysin tuloksena voidaan usein määritellä, koska tietoturvapoikkeama tapahtui, miten se tapahtui ja millaiset seuraamukset tietoturvapoikkeamalla oli. Vähintäänkin lokit voivat osoittaa seuraavaan tutkinnan kohteeseen, joka on mahdollisesti uutta vielä käsittelemätöntä lokia.

Lokianalyysiä tehdään tietoturvamielessä usein myös ilman havaittua tietoturvapoikkeamaa. Tällöin pyritään etsimään suuresta määrästä olemassa olevaa dataa anomalioita, jotka voisivat viitata väärinkäytöksiin verkossa (Crowdstrike, 2019). Yksi esimerkki tästä voisi olla esimerkiksi se, että analysoidaan käyttäjien kirjautumisia lähdemaan perusteella ja havaitaan,

että käyttäjä on kirjautunut tunnin sisällä sekä Suomesta ja Kanadasta. Tällöin kyseessä on anomalia. Tämän tyyppistä lokianalyysia kutsutaan tietoturva-alalla usein termillä Threat Hunting, uhkajahti.

3.2 Lokivalvonta

Lokivalvonta ja lokianalyysi ovat termeinä melko lähellä toisiaan, mutta tarkoittavat kuitenkin eri asiaa. Lokivalvonnalla tarkoitetaan kerätyn lokin valvomista. Lokivalvonta tuottaa yleensä automaattisesti hälytyksiä sääntöjen perusteella, esimerkiksi jos valvotaan palvelinten suorituskykyä, niin lokivalvonta hälyttää, jos jonkin palvelimen suorituskyvyssä havaitaan ongelma (Logdna). Tietoturvapuolella lokivalvonnalla voidaan hälyttää tietoturvapoikkeamasta, esimerkiksi siitä, että laitteella on havaittu haittaohjelma.

Lokivalvonnassa paljastuu usein anomalioita datasta. Verkkodatan valvonnassa voidaan havaita anomalioita liikenteen määrässä. Mikäli liikennemäärä nousee yhtäkkiä, tästä voidaan hälyttää. (GadAllag 2004, 10.) Tällaisella valvonnalla voidaan myös havaita haittaohjelmia. Madot esimerkiksi tekevät usein verkkoskannauksia, joita ei tehdä kovin usein normaalisti verkkoympäristöissä. Verkkovalvonta voi havaita tällaisen lokien perusteella ja tämän jälkeen haittaohjelma voidaan poistaa verkosta. (GadAllag 2004, 13.)

Lokivalvonta on käytännössä lokien reaaliaikaista valvontaa, eli lokeja valvotaan heti kun ne saapuvat valvovaan järjestelmään. Järjestelmä tuottaa hälytyksiä tästä datasta ja lähettää hälytykset määritellyllä tavalla määritetyille tahoille. Lokianalyysi on yleensä manuaalista ja sillä pyritään selvittämään jokin asia. Usein lokianalyysilla pyritään selvittämään esimerkiksi suorituskykyongelmia, tai mahdollisia ongelmia sovelluksessa. (Fitzpatrick.)

Lokivalvonnassa käytetään yleensä tätä toimenpidettä varten erikseen rakennettuja järjestelmiä, jotka pystyvät tehokkaasti normalisoimaan ja korreloimaan dataa useista eri lähteistä. Tätä dataa hyväksikäytetään automaattisella analyysilla ja tiettyjen ehtojen täyttyessä nostetaan tietoturvahälytys, joka ohjataan käsittelyyn. (Dunham 2019.)

Lokivalvonta tarkoittaa olemassa olevien lokilähteiden valvomista. Termillä tarkoitetaan jatkuvaa valvomista, eikä niinkään yksittäisessä tapauksessa tapahtuvaa valvontaa, joka menisi enemmän lokianalyysi termin alle. Tietoturvan näkökulmasta lokivalvonnan tarkoitus on valvoa, että tapahtuuko lokilähteissä asioita, joita siellä ei pitäisi tapahtua. (Dunham 2019.)

Lokianalyysissä mainittiin esimerkki, jossa etsittiin käyttäjän mahdotonta kirjautumista jossa, käyttäjä kirjautuu tunnin sisällä Suomesta ja Kanadasta. Lokivalvonnan ollessa kyseessä tällainen tapahtuma voisi aiheuttaa hälytyksen automaattisesti.

Lokivalvonnan käyttötapauksen tarkoitus on integroida lokilähteet keskitettyyn järjestelmään. Tämä keskitetty järjestelmä hoitaa sitten lokien analysoinnin ja korreloinnin, sekä tuottaa

hälytyksiä hälytyssääntöjen mukaisesti. Lokimäärät ovat todella suuria ja yksittäisten lokitapahtumien määrä voi nousta organisaatioissa miljooniin, tai miljardeihin. Tällöin ihmisen tekemä manuaalinen analyysi ei enää riitä, vaan tarvitaan automaatiota. Lokivalvontaa tehdään nykyään todella paljon suurissa yrityksissä, koska se auttaa havaitsemaan mahdolliset tietomurrot ympäristöissä. (Miller 2019.)

Yleensä tällaiset järjestelmät, jotka tekevät tietoturvavalvontaa, tunnetaan paremmin nimellä SIEM, eli Security Information and Event Management. (Dunham 2019.) ELK järjestelmä ei itsessään ole SIEM järjestelmä, mutta monet sen toiminnallisuudet ovat kyllä hyvin vastaavia. ELK järjestelmää voidaan käyttää myös SIEM järjestelmänä suoraan, mutta se vaatii erillisen lisensoidun paketin ostamisen valmistajalta. (Elastic.)

4 Tutkimus- ja kehittämismenetelmät

Tämä opinnäytetyö on luonteeltaan kehittämistyö. Opinnäytetyössä kehitetään kotiverkkoon lokivalvontajärjestelmä, joka valvoo kotiverkon tietoturvaa. Työ koskee yksittäistä järjestelmää ja sitä miten kyseinen järjestelmä pystyy suorittamaan lokien analysointia ja valvontaa ja järjestelmä toteutetaan valvomaan kotiverkkoa opinnäytetyötä tehdessä.

Osallistuva havainnointi sopii yhdeksi tämän opinnäytetyön aineistonkeruu menetelmäksi, koska tutkija asentaa ELK-järjestelmän. Tällöin tutkija osallistuu tutkimukseen aktiivisesti ja samalla havainnoi tutkimuksen kohdetta ja tekee tästä johtopäätöksiä. Osallistuvassa havainnoinnissa tutkija osallistuu aktiivisesti tutkittavaan kohteeseen. Tutkija ei kuitenkaan aktiivisesti pyri vaikuttamaan tutkimuksen kohteeseen, vaan pyrkii välttämään tätä. Tutkija pyrkii välttämään lopputuloksiin vaikuttamista, joskin tämä ei aina täysin onnistu. (Grönfors & Vilkka 2011, 52.)

4.1 Kehittämistutkimus

Kehittämistutkimuksessa yhdistyy kaksi asiaa, jotka ovat tutkimus ja kehitys. Kehittämistutkimus ei seuraa täysin samaa mallia kuin tieteellinen tutkimus, joka noudattaa aina tiettyjä sääntöjä. Kehittämistutkimuksessa sisältää myös tieteellisen tutkimuksen ominaisuuksia, mutta ne eivät vastaa täysin tieteellisen tutkimuksen menetelmiä. Tieteellinen tutkimus tuottaa tietoa, jonka perusteella voidaan aloittaa kehitys. (Salonen 2014, 9-10.)

Kehittämistutkimuksella on monia erilaisia kehittämismenetelmiä, jotka voivat poiketa jossain määrin määrällisestä ja laadullisesta tutkimuksesta. Kehittämistutkimuksessa käytetään usein menetelmänä esimerkiksi käytännön testausta, SWOT-analyysiä, työpajoja, benchmarkingia ja menetelmäkirjallisuutta. Aineistonkeruumenetelmänä voidaan myös käyttää useita erilaisia menetelmiä kuten havainnointia, kyselyjä ja valmista materiaalia. Tässä opinnäytetyössä kehittämismenetelmänä SWOT-analyysiä, käytännön testausta ja

menetelmäkirjallisuutta. Aineistonkeruumenetelmänä käytetään osallistuvaa havainnointia. (Salonen 2013, 22-23.)

Opinnäytetyön kehittämisvaiheessa pyritään dokumentoimaan tuotetut materiaalit tarpeellisella tasolla. Varsinaista asennusohjetta opinnäytetyöstä ei haluta, eli asennusta ei tulla dokumentoimaan kokonaisuutena, vaan enemmänkin asennuksen jälkeiset konfiguroinnit ja muutokset. Tällä pyritään varmistamaan, että kehittämistyö on dokumentoitu tarvittavalla tasolla, että siitä voidaan ymmärtää kehittämistyön etenemisen vaiheet. (Salonen 2013, 23-24.)

4.2 Käytännön toteutus ja testaus

Opinnäyteyössä tutkittava ELK-järjestelmä asennetaan tuotantoympäristöä vastaavaan käyttöön ja järjestelmään integroidaan oikeaa lokidataa. Järjestelmä tulee valvomaan oikeaa dataa, joskin data tullaan integroimaan normaalikäyttäjän kotiverkosta, joita kyseisessä kotitaloudessa on käytössä. Data vastaa kuitenkin hyvin monilta osin yritysverkkoa valvottavan datan osalta.

Tällä käytännön testillä voidaan havaita, kuinka hyvin järjestelmällä voidaan tehdä lokidatan analysointia ja valvontaa. Lisäksi käytännön testi selvittää kuinka paljon työtä tarvitsee tehdä, että järjestelmästä saadaan käyttökelpoinen. Käytännön testi pyrkii kuvaamaan miten lokianalyysia ja valvontaa voidaan suorittaa ELK-järjestelmällä.

4.3 Lähdekirjallisuus

Opinnäytetyön aiheesta löytyy suuri määrä valmista aineistoa, jotka tukevat hyvin työn toteuttamista. Näitä valmiita aineistoja analysoidaan ja niitä käytetään opinnäytetyön lähteenä. Valmiit aineistot, joita opinnäytetyössä käytetään ovat pääosin sähköisessä muodossa olevat artikkelit, sekä tietenkin järjestelmän valmistajan itse tuottama dokumentaatio. Opinnäyteyössä tutkittava järjestelmä päivittyy jatkuvasti, useita kertoja vuodessa, jonka takia painettu materiaali ei pysy päivitysten perässä. Opinnäytetyössä käytetään kuitenkin myös painettua materiaalia, mutta sen osuus on vähäinen.

Lähdekritiikillä pyritään varmistamaan, että opinnäytetyössä käytettävä lähdekirjallisuus on laadukasta ja sisältää totuuksia. Lähdekritiikkiä pitää aina arvioida tapauskohtaiseksi, koska se ei ole tekninen prosessi. Tutkijan täytyy miettiä itse, että onko käytettävä lähde totuudenmukainen. Mikäli käytetty lähde herättää epäilyksiä, niin kannattaa lähdettä verrata muihin lähteisiin, mikäli tällaisia on saatavilla. Tässä työssä käytetään suurimmaksi osaksi sähköisiä lähteitä, joten lähteitä on yleensä useita ja mikäli ensimmäinen lähde epäilyttää, niin voidaan oikeellisuus tarkistaa hyödyntäen useita lähteitä. (Alasuutari 2011, 71 - 75.)

Lähdekirjallisuuden valinnassa pyrittiin varmistamaan, että kirjallisuus on laadukasta. Lähteissä suosittiin tieteellisiä tutkimuksia, mikäli niitä oli saatavilla. Mikäli ei ollut, niin ensisijaisesti käytettiin järjestelmän valmistajan omaa materiaalia. Kaupallisten toimijoiden materiaali pyrittiin välttämään, mutta joissain tapauksissa myös kaupallisten tahojen materiaalia käytettiin, mikäli materiaali oli tutkimusluontoista eikä pyrkinyt edistämään tuotteen myyntiä. Lisäksi lähteissä huomioitiin niiden julkaisusivu ja epäilystä herättäviä lähteitä vältettiin. Opinnäyteyössä käytetään paljon blogitekstejä ja manuaaleja lähteenä, koska nämä tuottavat yleensä tuoretta tietoa, joka on oleellista nopeasti muuttuvan ELK-järjestelmän osalta.

4.4 SWOT-analyysi

SWOT-analyysi muodostuu sanoista Strenghts (Vahvuudet), Weaknesses (Heikkoudet), Objectives (Mahdollisuudet) ja Threats (Uhat). Analyysissa pyritään saamaan jokaiseen kenttään analyysin kohteesta vähintään yksi asia ja analyysi visualisoidaan yleensä nelikenttämuodossa. Vahvuudet ja heikkoudet tulevat yleensä analyysin kohteesta itsestään, eli ne ovat sisäisiä arvioita. Mahdollisuudet ja uhat ovat molemmat ulkoisia, eli jokin muu kuin analyysin kohde itsessään vaikuttaa näihin. Ulkoisia mahdollisuuksia voisi luoda esimerkiksi hyvät joukkoliikenneyhteydet, mikäli analysoitaisiin kauppakeskuksen mahdollista sijaintia. Uhka samassa esimerkissä voisi olla esimerkiksi viranomaisten haluttomuus antaa rakennuslupaa kyseiseen sijaintiin. (Vuorinen 2013, 64.)

Tuotteen testauksen perusteella luodaan SWOT-analyysi, jossa analysoidaan tuotteen lokianalysoinnin ja valvonnan vahvuuksia, heikkouksia, mahdollisuuksia ja uhkia. Analyysillä pyritään osoittamaan käytännönläheisesti hyviä ja huonoja puolia, joita tuotteessa havaittiin.

4.5 Sisällönanalyysi

Sisällönanalyysi liittyy yleensä laadulliseen tutkimukseen, mutta sen ollessa erittäin monipuolinen menetelmä voidaan sitä hyödyntää hyvin myös kehittämistyössä (Sarajärvi & Tuomi 2009,78). Sisällönanalyysin tarkoitus on analysoida lähes mitä vain dataa oli se sitten artikkeli tai haastattelu ja sillä on tarkoituksena tuottaa tuloksia tiivistetyssä muodossa käsitellystä datasta. Perimmäinen tarkoitus on analysoida materiaalin sisältöä ja tarkoitusta, eli pyrkiä ymmärtämään materiaalin tarkoitus. (Sarajärvi & Tuomi 2009, 88.)

Tässä työssä kerätty aineisto perustuu suurelta osin dataan, jota ELK-järjestelmän asennus tuottaa. Tästä datasta pyritään löytämään tarkoitus, sekä vetämään data yhteen. Tämän jälkeen datasta tehdään johtopäätöksiä.

4.6 Validiteetti ja reliabiliteetti

Reliabiliteetilla tutkimuksessa tarkoitetaan sitä, että tutkimuksen tulokset tulisi olla toistettavissa. Mikäli toinen tutkija tekee vastaavaa tutkimusta vastaavissa olosuhteissa, niin tutkimuksen tulokset pitäisi olla molemmissa tutkimuksissa vastaavat. Tällä voidaan vahvistaa löydökset, sekä varmistaa, että tutkimuksen ovat hyväksyttäviä. Tutkimusvälineiden tulee tukea tätä, eli välineiden tulee olla luotettavia. Lisäksi tuloksia mitatessa, tulee mittaus suorittaa useamman kerran, että lopputulosta voidaan pitää luotettavana. Mikäli tutkimusta ei voida toistaa, tai se on vaikea toistaa, niin tutkimustulos on vähemmän luotettava. (Vilkka 2015, 124.)

Validiteetilla määritellään, että kuvastavatko tutkimuksen lopputulokset sitä mitä niiden pitäisi. Reliabiliteetilla määritellään, että tulos on toistettavissa, mutta validiteetilla korostetaan nimenomaan lopputuloksen oikeellisuutta. Reliabiliteetti ei välttämättä määrittele, että lopputulos on järkevä vaan pelkästään sen, että lopputulos voidaan toistaa. Tällöin tutkimus voi olla toistettavissa, eli reliabiliteetti täyttyy, mutta tutkimuksen lopputulos voi silti olla sisällöltään väärä. Esimerkkinä tässä voisi olla väite, että pidemmät ihmiset ovat älykkäämpiä. Ihmisten pituuden mittaaminen on toistettavissa luotettavasti, mutta tutkimuksen lopputulos ei ole järkevä. Tutkimuksen reliabiliteetti täyttyy, mutta validiteetti ei. (Vilkka 2015, 124.)

5 ELK-järjestelmän käyttöönotto

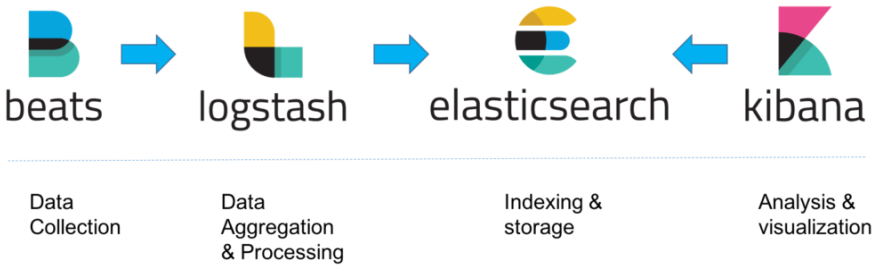
ELK-järjestelmän käyttöönotto aloitetaan valitsemalla opinnäytetyön kannalta soveltuvin ELK-järjestelmään pohjautuva sovellus. Opinnäytetyön kannalta oleelliset ominaisuudet analysoidaan ja näiden perusteella tehdään johtopäätös järjestelmästä, joka sopii parhaiten tämän työn puitteisiin. Analysoitavat ominaisuudet ovat järjestelmän soveltuvuus kotiverkon valvontaan, tuotteen hinta, sekä tuotteen muokattavuus opinnäytetyössä integroitaviin lokilähteisiin. Tämän jälkeen aloitetaan varsinainen järjestelmän asennus, lokilähteiden integrointi ja viimeisenä tehdään varsinaiset haut ja visualisoinnit, jotka mahdollistavat lokianalyysin ja valvonnan.

5.1 ELK -järjestelmän kuvaus

ELK-järjestelmä on vapaan lähdekoodin sovellus ja akronyymi termeistä Elasticsearch, Logstash ja Kibana. ELK tunnetaan usein termillä ELK Stack, johtuen siitä, että kokonaisuus koostuu useista eri tavallaan toisista irrallisista komponenteista. ELK kokonaisuuteen on nykyään liitetty myös uusi komponentti, joka tunnetaan nimellä Beats (Berman 2019).

Järjestelmän koostuessa useista komponenteista, käydään komponentit yksitellen läpi. Kuvio 2 kuitenkin kuvastaa miten komponentit nitoutuvat yhteen. Beats kerää dataa ja lähettää sen Logstashin käsiteltäväksi, josta se lähetetään edelleen Elasticsearch -tietokantaan. Kibana on

graafinen web -pohjainen käyttöliittymä, joka lukee dataa Elasticsearch tietokannasta ja Kibanalla voidaan luoda datan perustella analyyseja ja visualisointeja.



Kuvio 2: ELK järjestelmän komponentit. (Berman 2019.)

5.1.1 Beats

Beats on kokoelma erillisiä agentteja ja niiden tarkoitus on kerätä dataa Elasticsearch tietokantaan. Uusia räätälöityjä Beatseja tehdään koko ajan lisää ja niitä on suhteellisen yksinkertaista tehdä erilaiselle lokille (Objectrocket). Lokit ovat usein täysin erilaisia riippuen lähdejärjestelmästä ja Beatsien tarkoitus on käsitellä eri järjestelmien tuottamaa erilaista lokia, tai muuta dataa, jota halutaan analysoida.

Tällä hetkellä virallisia Beatseja, jotka valmistaja on luonut, löytyy seitsemän kappaletta (Elastic.). Valmistajan omat Beatsit ovat avointa lähdekoodia ja täten näistä löytyy erilaisia organisaatioiden ja henkilöiden räätälöimiä ratkaisuja, mutta Beatsien perimmäinen tarkoitus pysyy samana näissä kaikissa. Perimmäinen tarkoitus on käsitellä erilaista lokitietoa ja lähettää se Elasticsearch kantaan, tai Logstashiin. (Objectrocket.)

Nimi	**Tarkoitus**
Filebeat	Filebeat käsittelee levyltä löytyviä lokitiedostoja.
Packetbeat	Packetbeat käsittelee järjestelmien välistä verkkoliikennettä.
Metricbeat	Metricbeat valvoo käyttöjärjestelmien resurssinkäyttöä, kuten CPU-käyttöä.
Winlogbeat	Winlogbeat käsittelee Windows käyttöjärjestelmän tuottamaa lokitietoa.
Auditbeat	Linux käyttöjärjestelmien käyttäjätapahtumien lokitusta varten luotu Beat.
Heartbeat	Heartbeat valvoo palveluita ja palvelimia.
Functionbeat	Functionbeat kerää dataa pilvipalveluista.

Taulukko 1: ELK Beatsien käyttötarkoitus. (Elasticsearch.)

5.1.2 Logstash

Logstashia käytetään datan vastaanottamiseen erilaisista lähteistä ja Logstashilla voidaan muuttaa dataa ennen kuin se lähetetään eteenpäin Elasticsearch tietokantaan. Logstashilla voidaan muokata datan sisältöä, jota se ottaa vastaan, sekä muokata dataa niin, että se on koneluettavassa muodossa. Logstashilla voidaan myös rikastaa dataa ennen kuin se toimitetaan eteenpäin jatkokäsittelyyn. (Guru99.)

Beatsit ovat yksi esimerkki lähteestä, joka voi lähettää dataa Logstashiin, joskin muitakin lähteitä on. Beats on kuitenkin yleinen lähde, jolla dataa syötetään Logstashiin. Lisäksi Logstashisissa on liitännäisiä, joiden kautta järjestelmään voidaan syöttää dataa. Tällaisia liitännäisiä ovat esimerkiksi syslog ja twitter, jolloin Logstashiin voidaan lähettää suoraan syslog ja twitter dataa (Elastic).

5.1.3 Elasticsearch

Elasticsearch on Javaan perustuva tietokanta, jonka tarkoitus on mahdollistaa tekstihaut tietokannassa olevaan dataan. Tämä tarkoittaa sitä, että kaikki kannassa oleva tieto on helposti haettavissa, eikä pelkästään metadata, kuten joissain tietokantamalleissa.

Tietokannan sisältämä data on indeksoitua dataa, joka tuotetaan Apache Lucene tuotteella. Lucene on todella monimutkainen järjestelmä, mutta Elasticsearch tarjoaa erilaisia API-rajapintoja, joten Elasticsearchin käyttäjän ei tarvitse välttämättä osata käyttää Lucenea laisinkaan. Elasticsearchn käyttötarkoitus on tallentaa ja säilöä suurta määrää erilaista dataa ja tarjota erittäin nopeasti toimiva hakutoiminto tämän datan analysoimista varten. Tietokanta käyttää JSON formaattia. (Chand 2019.)

Elastisearchin avainkomponentteja on muutamia. Yksi niistä on indeksi, joka sisältää loogista dataa joka järjestelmään tuodaan. Indeksit mahdollistavat nopeat haut tietokannan sisältämään dataan, mutta indeksit itsessään ei sisällä alkuperäistä dataa. Tietokannan sisällä säilöttävät datat ovat pääosin nimeltään dokumentteja. Nämä dokumentit sisältävät kenttiä ja sama kenttä voi esiintyä useita kertoja yhdessä dokumentissa, jolloin kyseessä on useita arvoja sisältävä kenttä. Arvoilla on aina jokin tyyppi, kuten päivämäärä, teksti, numero tai muu vastaava. Nämä dokumentit ovat muodoltaan JSON formaatissa. Elasticsearchin sisältämä data on voitu jakaa useille palvelimille. (Kuc & Rogozinski 2014, 41-42.)

Yhdessä Elasticsearchin indeksissä voi olla useita erilaisia objekteja, joilla on erilainen tarkoitus. Nämä erilaiset objektit erotellaan dokumentin tyyppi kentällä. Vaatimuksena on kuitenkin se, että kenttien arvot pitää pysyä kaikissa dokumenteissa samana tyypistä riippumatta, eli esimerkiksi Päivämäärä niminen kenttä sisältää aina saman tyypin dataa. Dokumentin eri kentät vaativat erilaista analysointia. Elasticsearch käyttää kartoitusta tämän tyypin

määrittelyyn ja kartoituksessa on määritelty datatyyppi jokaiselle kentälle. (Kuc & Rogozinski 2014, 42.)

5.1.4 Kibana

Kibana on selainpohjainen sovellus, joka lukea dataa Elasticsearch tietokannasta. Kibanalla voidaan tehdä helposti graafisen käyttöliittymän kautta hakuja indeksoitua dataa vasten, sekä luoda erilaisia visualisointeja datasta. Kibanan yksi suurimpia vahvuuksia on sen keinot pystyä luomaan helposti lähestyttäviä visualisointeja, sekä kokoelmia visualisoinneista, jolloin voidaan yhdeltä ruudulta nähdä suurikin määrä visualisoitua dataa suhteellisen yksinkertaisessa muodossa. (Berman, 2019.)

Kibanan hakutoiminnallisuus tukee vapaatekstihakua, jolloin voidaan hakea haluttua dataa koko datamassasta, jota tietokannassa säilötään. Lisäksi voidaan tehdä rajoittuneempia hakuja, jolloin voidaan esimerkiksi määritellä tietyt kentät, josta dataa haetaan pelkästään. Lisäksi jo tehdyt haut voidaan tallentaa omiksi näkymikseen, joihin voidaan määritellä taulukkomuotoisena, että mitkä datassa olevat kentät näytetään käyttäjälle. (Berman, 2019.)

5.2 Asennettavan ELK järjestelmän valinta

Aikaisemmin opinnäytetyössä mainittiin, että ELK on hyvin yleisessä käytössä tällä hetkellä jo, osittain sen takia, että ELK pohjautuu avoimeen lähdekoodin. Tämän takia useat eri organisaatiot ja henkilöt ovat aloittaneet työstämään valmista lokianalyysijärjestelmää käyttäen taustalla hyväkseen ELK:a. Tällöin ei ole välttämättä järkevää käyttää suoraan puhdasta ELK:a, koska jossain muussa järjestelmässä saattaa olla valmiiksi tehtynä suuri osa tarvittavista integraatioissa, visualisoinneista ja hauista.

Lokilähteitä integroitaessa on tärkeää kuitenkin, että valittava järjestelmä on hyvin räätälöitävissä. Mikään järjestelmistä ei tule tukemaan kaikkia niitä lokilähteitä, joita opinnäytetyössä tullaan tarvitsemaan. Lisäksi opinnäytetyön kannalta on olennaista, että valittava järjestelmä on suhteellisen lähellä alkuperäistä ELK toteutusta tekniseltä toteutukseltaan. Täysin erilaiseksi räätälöity järjestelmä voi tehdä niin paljon muutoksia, että muutoksien tekeminen tähän järjestelmään on liian hankalaa tai aikaa vievää.

Erittäin toivottavaa lokilähteiden integraatioiden kannalta olisi, että erityisesti taustalla toimiva Elasticsearch-tietokanta olisi täysin valmistajan suositusten mukainen. Tällöin voidaan luoda vain yksi kanta, johon viedään käytännössä kaikki lokit, jota opinnäytetyössä tullaan käyttämään. Mikäli Elasticsearch on vahvasti räätälöity voi olla mahdollista, että uusien lokien vieminen ei ole lainkaan tuettua, tai sitten se poikkeaa niin paljon normaalista, että järjestelmää ei kannata ottaa käyttöön. Lisäksi valittavan järjestelmän tulee olla ilmainen.

5.2.1 ELK:in vahvuudet ja heikkoudet

ELK on käsitelty itsessään jo aikaisemmin tässä opinnäytetyössä, joten tämän otsikon tarkoitus on analysoida ELK:n hyviä ja huonoja puolia jos mietitään muita ELK:n pohjautuvia järjestelmiä, joita opinnäytetyössä käsitellään.

ELK järjestelmän valintaa opinnäytetyössä käytettäväksi alustaksi tukee se, että se on suoraan toimittajan tarkoittama tapa käyttää järjestelmää, ilman mitään lisäasetuksia. Tällöin tuote voidaan räätälöidä omaan käyttöön sopivasti suhteellisen helposti ja mahdollisiin ongelmiin löytyy todennäköisimmin ratkaisu suhteellisen yksinkertaisesti. Tuotteesta löytyy todella hyvät dokumentaatiot, jotka soveltuvat suoraan asennustilanteeseen. Lisäksi arkkitehtuurisesti ELK on kaikista joustavin valinta, koska järjestelmää ei ole räätälöity laisinkaan. Mikäli kuitenkin valitaan ELK, niin integroinnit, haut ja visualisoinnit joudutaan tekemään kokonaan uudelleen, joka voi lisätä työtä huomattavan paljon.

Hyvät puolet huomioiden järjestelmä käyttötavan ja toteutuksen:

- Arkkitehtuuri on hyvin tunnettu ja erilaiset vaihtoehdot tuettuja.
- Ongelmatilanteisiin löytyy todennäköisimmin helposti ratkaisu.
- Räätälöinti on helppoa, koska tuotetta ei ole etukäteen räätälöity ja muutettu.

Huonot puolet:

- Integroinnit, haut ja visualisoinnit pitää tehdä alusta lähtien kokonaan uusiksi. Tämä voi lisätä työmäärää huomattavan paljon.
- Hälytyssääntöjä ei välttämättä tueta laisinkaan.

5.2.2 SOF-ELK

SOF-ELK on Phil Hagenin luoma työkalu, joka on aikanaan tehty SANS organisaation kurssia varten, jossa Phil itse koulutti. Tämä järjestelmä on luotu nimenomaan tietoturvapoikkeamien selvittämistä varten ja sen perimmäinen tarkoitus on helpottaa lokianalyysia. Työkalu helpottaa tietoturva-asiantuntijoiden työtä, jotka ovat selvittämässä poikkeamia, koska SOF-ELK asentuu hyvin nopeasti. Asennusmedia on virtualisoitu applikaatio, eli se voidaan käynnistää lataamisen jälkeen suoraan virtualisointialustan päällä. (Hagen.)

SOF-ELK on kehitetty hallitsemaan useita eri lokityyppejä suoraan asennuksen jälkeen. Työkalu käyttää hyväkseen ELK:n mukana tulevia työkaluja, eli lokitiedostot käyttävät Filebeat työkalua, josta ne viedään Logstashiin ja Elasticsearchiin. Kibanaan on tehty valmiit

visualisoinnit ja yhdistetty näkymä, jonka kautta pystyy analysoimaan valmiiksi luotuja visualisointeja suoraan. (Hagen.)

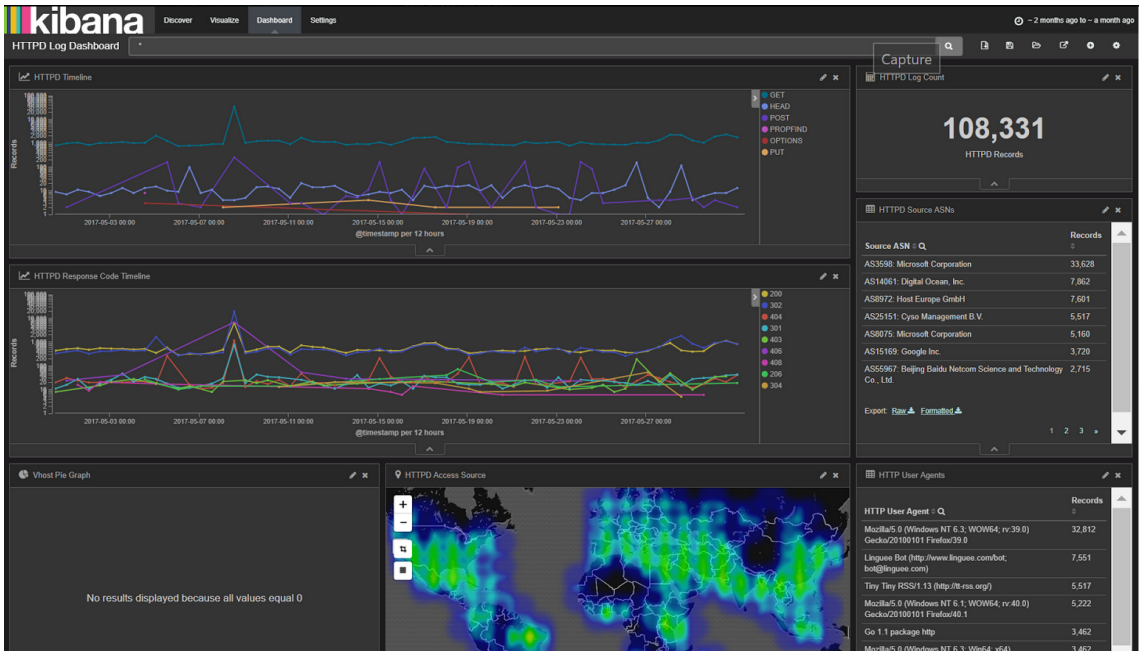
Lokit viedään SOF-ELK:n suoraan, eli virtuaalikoneeseen otetaan esimerkiksi SCP yhteys, jonka kautta lokit siirretään soveltuviin hakemistoihin. SOF-ELK tukee vain lokeja, jotka ovat tiedostoformaatissa ja jotka siirretään suoraan ennalta määriteltyihin kansioihin.

Seuraavat lokilähteet ovat suoraan tuettuna järjestelmässä:

- Syslog, yleinen lokiformaatti, jota monet eri järjestelmät, kuten Linux käyttöjärjestelmät käyttävät.
- Nfarch, NetFlow formaatti, eli nauhoitettua verkkodataa ilman varsinaista pakettidataa.
- Httpd, Apache lokit kolmessa eri muodossa.
- Passivedns, Passivedns sovelluksen lokit.
- Kape, Kape sovelluksen tuottama loki. Tämä on hyvin spesifinen loki, jota tietomurroissa käytetty sovellus tuottaa.
- Plaso, Plaso sovelluksen tuottama loki. Kuten edellinenkin tämä on todella spesifiseen tarpeeseen tehty sovellus.

(Hagen.)

SOF-ELK on todella hyvin toteutettu siihen käyttötarkoitukseen mihin se on tehty. Järjestelmä on käytännössä tarkoitettu nimenomaan tietomurtojen tutkimiseen, mutta ei niinkään jatkuvan tietoturvavalvontaan. Asennus on erittäin helppoa, mutta ei kovinkaan joustavaa räätälöintiin, joten komponentteja ei voi esimerkiksi erotella eri palvelimille. Teoriassa se olisi mahdollista, jos asennettaisiin useita SOF-ELK järjestelmiä, mutta tämä ratkaisu ei ole kovin puhdas. Kuvio 3 Kuvaa SOF-ELK:n valmista http -lokin visualisoitua näkymää, joka on sisäänrakennettu järjestelmään.



Kuvio 3: SOF-ELK httpd lokin visualisointinäkymä (Mcree).

SOF-ELK:n hyvät puolet verkonvalvonnassa:

- Tuotteessa on valmiina useita eri lokilähteitä. Kaikkea ei tarvitse välttämättä tehdä itse alusta.
- Valmiit visualisoinnit ja tallennetut haut ovat erittäin käyttökelpoisia.
- Tuotteen taustalla toimivaa ELK:a ei ole teknisesti muutettu hirveän paljon, joten tuotteen ominaisuudet eivät eroa juurikaan puhtaasta ELK asennuksesta.
- Tuote toimitetaan valmiina virtuaalipalvelimena, joka voidaan viedä suoraan virtualisointialustalle, joten asennus on erittäin helppoa.

Huonot puolet:

- Valmiiksi toimitettu virtuaalipalvelin aiheuttaa ongelmia, jos tuotetta halutaan räätälöidä esimerkiksi niin, että yksi ELK komponentti on eri palvelimella kuin toiset. Tämä on arkkitehtuurin kannalta erittäin huono asia.
- Lokilähteitä on suhteellisen paljon valmiina, mutta kaikki perustuu Filebeatiin, eli tiedostot pitää tuoda itse palvelimelle. Lisäksi suurin osa valmiista lokilähteistä ei ole tämän opinnäytetyön kannalta oleellisia.
- Tuote ei skaalaudu hyvin ja sitä ei ole tehty tietoturvavalvontaa varten, vaan tietoturvapoikkeamien tutkimista varten.

5.2.3 Security Onion

Security Onion on verkon valvontaan tehty ilmainen vapaan lähdekoodin sovellus. Järjestelmässä on nykyään huomattavan paljon ominaisuuksia ja sillä voi verkon lisäksi valvoa myös päätelaitteita. Tuote on todella laaja ja siitä löytyy toiminnallisuuksia useisiin erilaisin käyttötapauksiin. (Security Onion.)

Security Onion on koostettu useista erilaisista ilmaisen lähdekoodin järjestelmistä, ja tuotteen päätoiminnot ovat täysi pakettikaappaus (”Full Packet Capture”), Tunkeutumisen havaitsemisen järjestelmä verkkoon ja päätelaitteille, sekä analysointityökalut, joilla voidaan analysoida mahdollisia poikkeamia verkossa. Toiminnoissa käytetään paljon alalla hyväksi havaittuja tuotteita ja ne on nivottu yhteen hyvin toimivaksi kokonaisuudeksi. (Security Onion.)

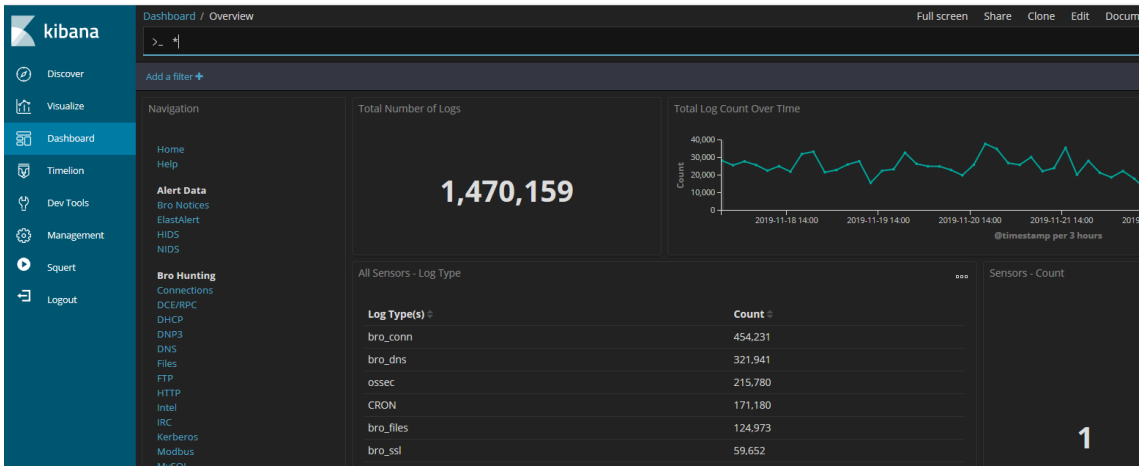
Tuote tukee useita erilaisia käyttötapauksia, jotka ovat listattuna seuraavaksi:

- Tuotteella voidaan tehdä verkkodatan analyysiä.
- Tuote voidaan asentaa tunkeilijan havaitsemisjärjestelmäksi erilaisissa arkkitehtuureissa.
- Tuote voidaan asentaa analysointikäyttöön niin, että dataa käsitellään palvelimella.
- Tuote voidaan asentaa niin sanotuksi sensoriksi, jolloin tuote analysoi sinne vietävää dataa ja lähettää hälytykset eteenpäin toiseen järjestelmään.

(Security Onion.)

Security Onionin arkkitehtuuri mahdollistaa asennuksen erilaisissa kombinaatioissa. Tuote voidaan asentaa kokonaisuudessa yhdelle palvelimelle, tai jakaa kuormaa useille eri palvelimille. Yksittäisen palvelimen asennusta ei suositella tuotantokäytössä laisinkaan, vaan tuote pitäisi asentaa aina kuormanjakomuodossa. (Security Onion.)

Tuote on erittäin varteenotettava vaihtoehto tämän opinnäytetyön kohteeksi valittavaksi järjestelmäksi. Tuote pystyy ottamaan vastaan useita eri lokilähteitä, jotka tullaan integroimaan opinnäytetyön puitteissa. Tämä helpottaisi työtä jonkin verran, kun kaikkia lähteitä ei tarvitse integroida alusta loppuun. Security Onion on kuitenkin vahvasti räätälöity ja tuotteessa on ainakin osittain muutettu myös ELK järjestelmän perimmäistä logiikkaa, joka voi tehdä muiden järjestelmien integroinnista hankalaa. Kuvio 4 esittää Security Onionin Kibanaan tuottamaa aloitusnäkymää.



Kuvio 4: Security Onionin Kibana näkymä. Kuva omasta ympäristöstä.

Security Onionin hyvät puolet verkonvalvonnassa:

- Tuotteessa on valtava määrä valmiita työkaluja, joilla voidaan tehdä analyysiä.
- Useille eri lokilähteille löytyy valmis integraatio.
- Tuote tukee jaettua arkkitehtuuria, jossa tietokannat on jaettu omalle palvelimelle.

Huonot puolet:

- Taustalla toimivaa ELK järjestelmää on räätälöity, hankaloittaa uusien lokien konfigurointia.
- Tuote on tehty lähtökohtaisesti valvontaa varten, ei analysointia varten.
- Tuote tuottaa tiettävästi laadukkaita hälytyksiä, mutta vain jos säännöt ovat konfiguroitu hyvin.

5.2.4 Logz.io

Logz.io on melko hyvin tunnettu ja laajassa käytössä oleva järjestelmä. Tuote on esillä opinnäytetyössä lähinnä sen takia, että se on niinkin tunnettu kuin on, mutta sen suurempaa analyysiä tuotteen kyvykkyyksistä ei tehdä johtuen siitä, että tuotteen ilmainen versio on liian rajattu ja se ei itsessään sovi opinnäytetyöhön. Tuotteesta tarvittaisiin maksullinen versio tähän opinnäytetyöhön, joten tuote rajataan suoraan pois tämän takia.

Ilmaisen version rajoitteet ovat seuraavat: 3GB datan käsittelyä päivässä, 3 päivän datan säilytysaika, 5 käyttäjää ja 50 hälytystä (Logz.io 2019). Näistä rajoitteista erityisesti liian rajattu datan käsittelymäärä ja datan säilytysaika sulkevat vaihtoehdon kokonaan pois.

5.2.5 Yhteenveto ja valinnan perustelu

Järjestelmiä, jotka käyttävät ELK järjestelmää tietoturvalokien analyysiin ja valvontaan on useita. Tässä opinnäytetyössä käsiteltiin muutamia vaihtoehtoja, jotka voisivat toimia opinnäytetyöhön valittavana järjestelmänä. Kaikissa vaihtoehdoissa oli hyviä ja huonoja puolia, joten valinta ei ollut helppoa. Taulukko 2 kuvaa työn kannalta oleelliset asiat, joiden perusteella valinta suoritettiin.

	Logz.io	**Security Onion**	**ELK**	**SOF-ELK**
Vaatii lisenssin	Kyllä	Ei	Ei	Ei
Tuote on räätälöity ja ei vastaa integroinneilta puhdasta ELK-järjestelmää	Ei	Kyllä	Ei	Ei
Tuotteessa on luotu valmiita visualisointeja ja hakuja	Kyllä	Kyllä	Ei	Kyllä
Arkkitehtuuri on eriytettävissä useille palvelimille	Kyllä	Kyllä	Kyllä	Ei

Taulukko 2: ELK-järjestelmien opinnäytetyön kannalta tärkeät ominaisuudet.

Taulukossa on muutama kohta, joista ensimmäinen on yksiselitteinen. Opinnäytetyössä ei voida käyttää tuotetta, joka vaatii lisenssin, koska kotiverkonvalvontaan ei haluta käyttää suurta summaa rahaa.

Toinen kohta ”Tuote on räätälöity ja ei vastaa integroinneilta puhdasta ELK-järjestelmää”, tarkoittaa, että tuotteen pohjalla olevaa ELK-järjestelmää on muutettu niin, että lokilähteiden integroinneissa käytetään puhtaasta ELK-järjestelmästä poikkeavia tapoja. Tällöin lokilähteiden integraatio voi olla huomattavasti työläämpää.

Kolmannella kohdalla ”Tuotteessa on luotu valmiita visualisointeja ja hakuja” tarkoitetaan, että tuotteeseen on tehty valmiiksi näkymiä, jotka avustavat analyysin tekemisessä. Tämä on tuotteelta toivottavaa, koska se vähentää työmäärää, kun kaikkia näitä visualisointeja ei tarvitse tehdä alusta asti itse.

Neljäs kohta, eli ”Arkkitehtuuri on eriytettävissä useille palvelimille” tarkoittaa sitä, että tuotteen arkkitehtuuri mahdollistaa roolien jaon eri palvelimille. Tämä mahdollistaa kuormanjakamista tasaisemmin eri palvelimille, jolloin kaikki kuorma ei kohdistu yhteen laitteeseen.

Logz.io suljettiin pois suoraan, johtuen valitettavasta lisenssivaatimuksesta. Ilmainen versio on sen verran rajoittunut, että se ei aja asiaa tämän opinnäytetyön puitteissa. Ilmainen versio soveltuu ihan hyvin testikäyttöön, mutta tässä opinnäytetyössä toteutettava järjestelmä valvoo oikeata dataa, joskin data on rajattu lähinnä kotikäyttäjän käyttämien järjestelmien sisältävään lokiin ja hälytyksiin.

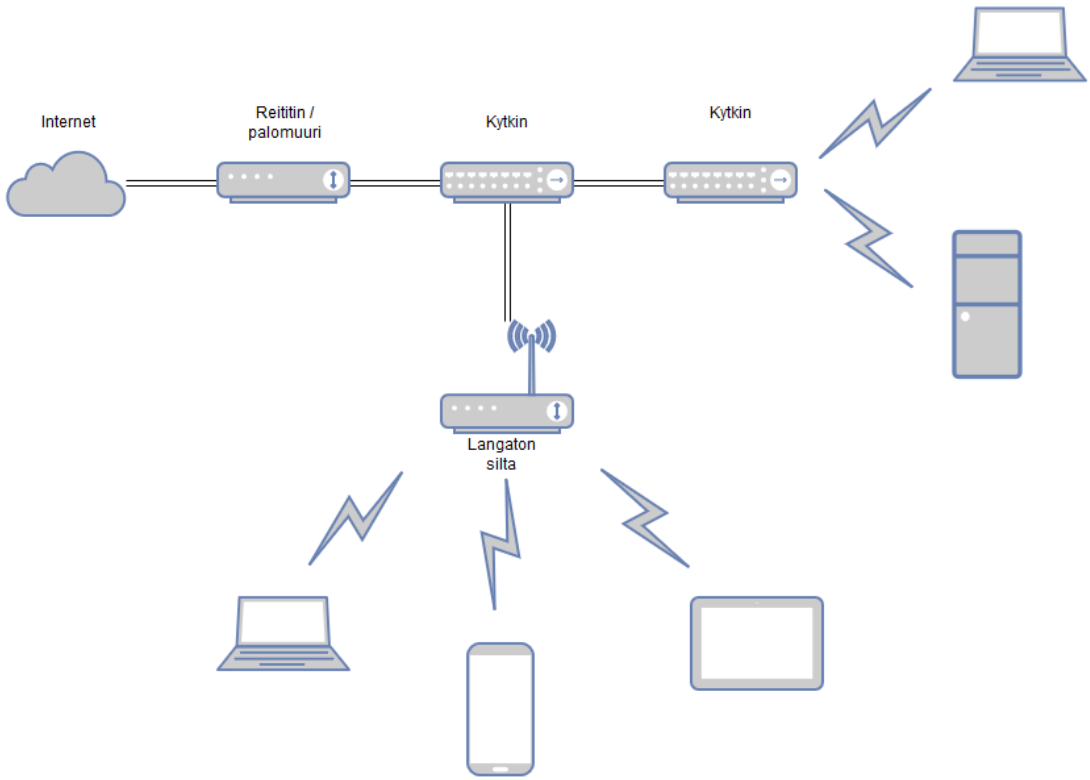
Security Onion olisi muuten oikein hyvä vaihtoehto, mutta koska tuotteessa on hyvin vahvasti räätälöity versio ELK:a pohjalla, niin tuotteessa olisi liian suuri riski siihen, että kaikkia haluttua ei saataisi tuotua sovellukseen. Täten Security Onionia ei valita opinnäytetyössä käytettäväksi tuotteeksi, mutta se valitaan sensoriksi, joka tuottaa dataa ja hälytyksiä varsinaiseen valittuun järjestelmään.

SOF-ELK:n tuottamat valmiit visualisoinnit tuovat suuren lisäarvon mitä voitaisiin hyödyntää opinnäytetyössä, mutta lokien integroinnit on tehty täysin tiedostopohjaisesti. Tämä aiheuttaa haasteita, koska opinnäytetyö tulee tekemään aktiivista valvontaa, jolloin tiedostopohjainen valvonta ei toimi. Lisäksi SOF-ELK ei tue arkkitehtuuria, jossa tietokanta on eriytetty omalle palvelimelleen, mikä on ensisijainen tapa, jolla tuote halutaan opinnäytetyön puitteissa asentaa. SOF-ELK:a voidaan silti hyödyntää työssä niin, että tuotteesta otetaan soveltuvat osat omaan käyttöön, eli tietyt konfiguraatiot ja mahdollisesti osa visualisoinneista.

Tuotteista jäljelle jääkin puhdas ELK asennus. Tämä on opinnäytetyöhön paras vaihtoehto, jotta arkkitehtuuri saadaan vastaamaan tarvetta. Lisäksi puhtaan ELK asennuksen ilo on se, että järjestelmää ei ole räätälöity etukäteen, joka helpottaa uusien integrointien ja räätälöintien tekoa. Tuotteesta löytyy myös eniten dokumentointia ja apua, mikäli työssä kohdataan ongelmia.

5.3 Asennettavan ympäristön kuvaus ja käyttötarkoitus

Ympäristö johon ELK tullaan asentamaan vastaa suhteellisen normaalia kotiverkkoa. Verkossa on joitain verkkolaitteita, kuten reititin, kaksi kytkintä ja Wifi silta. Lisäksi verkossa on erilaisia päätelaitteita, kuten kannettavia tietokoneita, matkapuhelimia, älytelevisioita ja palvelimena toimiva työasema. Kuvio 5 kuvaa verkon arkkitehtuuria.



Kuvio 5: Valvottavan verkon arkkitehtuurikuva.

Verkosta löytyy suhteellisen monipuolisesti erilaisia laitteita, kuten älylaitteita, tietokoneita ja matkapuhelimia. Ympäristö on suhteellisen monipuolinen kotiverkoksi ja monilta osin vastaa myös tuotantokäytössä nähtäviä yritysverkkoja. Ympäristö luokin melko normaalilta näyttävän valvottavan ympäristön tietoturvanäkökulmasta, jos verrataan yritysverkkoihin, joskin toki huomattavan paljon pienemmässä mittakaavassa.

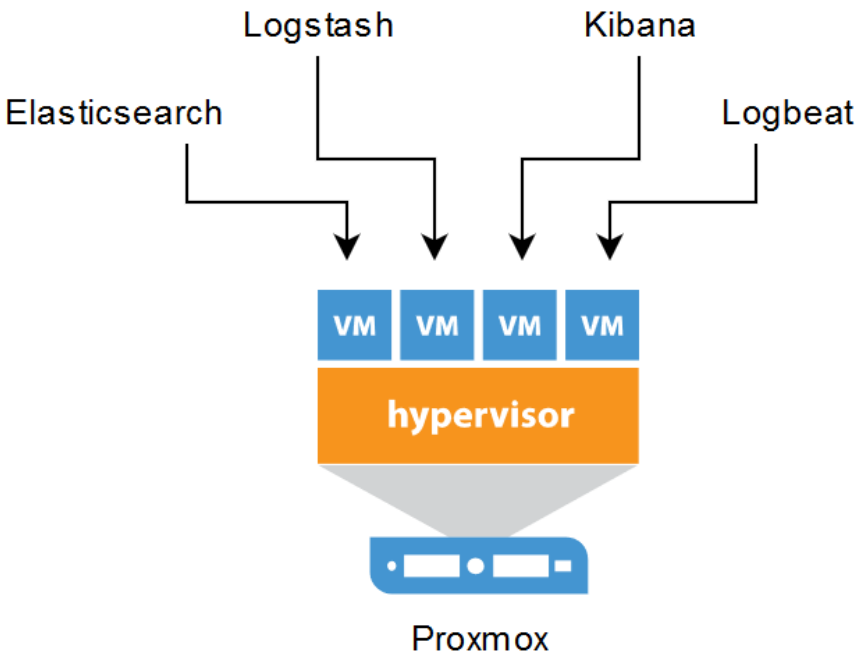
Käyttöönotettava ELK järjestelmä tulee keräämään erilaista dataa verkosta ja tarjoaa alustan, jolla voidaan tehdä valvontaa ja analyysiä tätä dataa vasten. Data on hyvin relevanttia tietoturvapoikkeamia selvittäessä ja dataa vasten voidaan myös tehdä valvontaa. Integroitavat lokit käydään myöhemmin opinnäytetyössä tarkemmin läpi, mutta ylätasolla listattuna data koostuu seuraavasta:

- Pääsynhallinnan lokeja useista järjestelmistä.
- Sovelluslokia useista järjestelmistä.
- Verkkodata kokonaisuutenaan, nauhoitettuna yhdeltä kytkimeltä.
- Hälytyksiä erilaisista tunkeutumista havaitsen järjestelmien toimesta.

5.4 Asennussuunnitelma ja asennus

Tuote joka opinnäytetyön toteuttamiseen valittiin, on ELK ja normaali ELK versio ilman mitään ulkoisia räätälöintejä. Tällöin ympäristö suunnitellaan tuotteen valmistajan parhaiden käytäntöjen mukaan miettien, miten komponentit kannattaa eriyttää toisistaan käyttötapauksen parhaan toimivuuden kantilta. Lisäksi ELK järjestelmä voidaan asentaa joko fyysiselle tai virtuaaliselle palvelimelle ja mikäli käytetään virtuaalista palvelinta voi palvelin olla myös konttipalvelin.

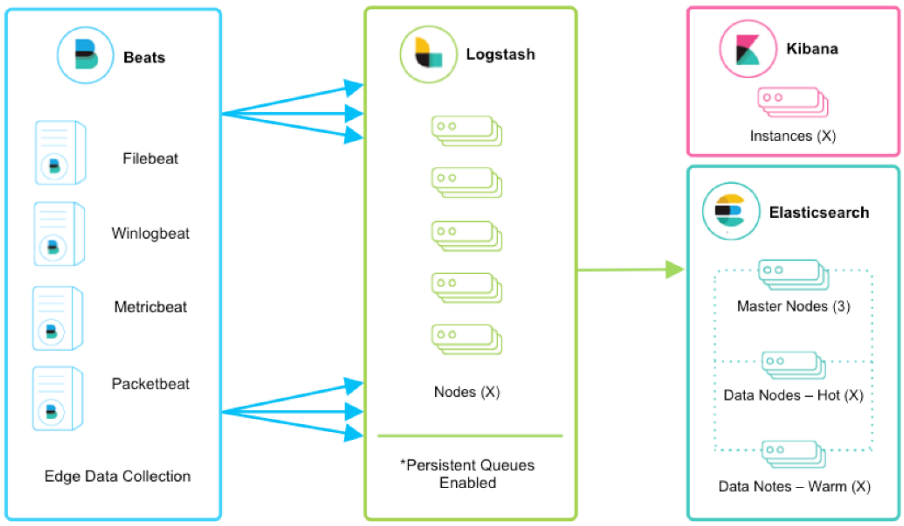
Asennettavasta verkosta löytyy tehotyöasema, joka sisältää Proxmox-nimisen virtualisointialustan ja tämä tulee toimimaan myös alustana, jolle ELK-järjestelmä asennetaan. Proxmox tukee niin kontti -pohjaisia palvelimia kuin normaaleja virtuaalipalvelimiakin. Opinnäytetyön luonteen takia konttipalvelin ei tuo mitään lisäarvoa ja itse ELK palvelin tuleekin toimimaan kiinteästi asennettuna, pitkäaikaisena toteutuksena, joten normaali virtuaalipalvelin on järkevämpi ratkaisu opinnäytetyöhön. Seuraavassa kuviossa 6 kuvataan ELK arkkitehtuuri, yksinkertaistettuna.



Kuvio 6: Yksinkertaistettu ELK arkkitehtuuri.

Opinnäytetyössä käytettiin useita eri tuotteita, jotka hyödyntävät Elasticsearch kantaa. Osa tuotteista kirjoittaa kantaan omaa dataansa, joka sitten analysoidaan muissa työkaluissa. Suunniteltuun käyttöön parhaiten soveltuu ratkaisu, jossa asennetaan yksittäinen Elasticsearch palvelin, joka tulee säilömään lähestulkoon kaiken lokidatan, jota opinnäytetyössä tallennetaan. Logstashille ja Kibanalle tehdään omat palvelimensa, jotka yhdistävät omilta

osiltaan keskitetylle Elasticsearch palvelimelle. Data, jota käsitellään ei ole kriittistä, joten kahdennusta ei käytetä. Kuvio 7 kuvaa käytettävää arkkitehtuuria, joskin Elasticsearchin ja Logstashin osalta nodeja ei ole useita, vaan vain 1 kumpaakin.



Kuvio 7: Käytettävä arkkitehtuuri. (Elastic.)

Asennus tapahtuu tuotteen valmistajan ohjeiden mukaisesti ja asennus itsessään on hyvinkin yksinkertainen toimenpide. Asennuksen jälkeen tapahtuva konfigurointi on asennuksen päävaihe, jotta järjestelmä saadaan toimimaan oikein yhteen. Jokainen virtuaalipalvelin asennetaan Ubuntu Server 18.04 käyttöjärjestelmä, koska se on erittäin helppokäyttöinen ja helppo asentaa. Ubuntu on myös vakaa ja usein päivittyvä käyttöjärjestelmä, joten se soveltuu hyvin myös käyttöön ELK:n kanssa. Taulukossa 3 kuvataan jokaiselle palvelimelle määritellyt resurssit, eli vCPU (Virtuaali prosessoriydin), RAM (keskusmuisti) ja levytyyppi ja palvelimelle määritelty levytila.

Rooli	vCPU	RAM	Levytyyppi ja tila
Elasticsearch	10	20GB	400GB SSD
Logstash	8	8GB	100GB SSD
Kibana	4	4GB	27GB SSD
Filebeat	4	8GB	80GB SSD

Taulukko 3: ELK asennuksen virtuaalipalvelinten resurssit rooleittain.

Kaikki palvelimet saavat erikseen nimen. Opinnäytetyössä tullaan käyttämään myös toimialuetta, joka on julkisesti rekisteröity osittain tätä käyttöä varten. Tämä mahdollistaa julkisten varmenteiden hankkimisen, mikäli sellaisia tarvitaan.

5.5 Järjestelmän asetukset

ELK järjestelmässä on useita asetuksia, jotka ovat erittäin merkitseviä tuotteen käytön kannalta. Järjestelmän asetuksia tulee muuttaa oletuksista, että tuote toimii mahdollisimman tehokkaasti ja osaa hyödyntää sille annettuja resursseja tarvittavissa määrin.

Lisäksi tuotteen asetuksista tulee miettiä esimerkiksi lokien kierrätystä, koska levytilaa on rajoitetusti ja tuotteen tulee poistaa tietokannasta automaattisesti lokeja vanhemmasta päästä. Tätä varten tulee olla soveltuva konfiguraatio, joka osaa hoitaa nämä poistot automaattisesti. Tuote asennetaan useille erillisille palvelimille, joten asetukset, joilla eri osat saadaan yhdistettyä toisiinsa, kuvataan myös.

5.5.1 Logstash asetukset

Logstash liitetään muuhun järjestelmään yksittäisten konfiguraatioiden osalta, eli käytännössä Logstash konfiguraatioon tehdään sääntöjä ja näissä säännöissä määritellään mihin järjestelmään käsitellyt lokitiedot lähetetään. Näin ollen, Logstashia itsesään ei liitetä keskitetysti asetuksissa muihin järjestelmiin. Konfiguroinnit tapahtuvat erikseen jokaisen integroidun lokilähteen osalta.

5.5.2 Elasticsearch asetukset

Elasticsearch on koko järjestelmässä eniten resursseja vaativa komponentti ja vastuussa järjestelmän tietokannasta. Täten Elasticsearch -palvelimelle onkin annettu eniten resursseja käyttöönsä. Elasticsearchin oletusasetuksia on myös muutettava, niin että palvelin käyttää sille määriteltyä IP -osoitetta. Ilman tätä määritystä palvelin kuuntelee vain 127.0.0.1 osoitetta ja ei ota vastaan kutsuja muualta.

Elasticsearchin yleiset asetukset tehdään konfiguraatiotiedostoon, joka on tässä asennuksessa polussa /etc/elasticsearch/elasticsearch.yml. Tämän konfiguraatiotiedoston sijainti riippuu siitä, että mikä on käyttöjärjestelmäversio ja millä tavalla tuote asennettiin. Tässä tapauksessa Elasticsearch asennettiin Ubuntu palvelimelle hyödyntäen valmistajan DEB pakettia ja apt hallintajärjestelmää.

Muutokset joita elasticsearch.yml tiedostoon tehtiin:

```
Network.host: 192.168.2.215
http.port: 9200
discovery.seed_hosts: ["192.168.2.215"]
```

Discovery.seed_hosts arvo on ainut, joka näistä ei ole yksiselitteinen. Tämä arvo tulee määritellä vastaamaan network.host arvoa asennuksessa jossa ei ole varsinaista Elasticsearch klusteria, vaan vain yksi palvelin joka vastaa koko palvelusta.

Elasticsearch tarvitsee asetukset, joiden perusteella vanhaa tietoa poistetaan järjestelmästä. Mikäli näitä asetuksia ei määritellä tulee levy täyttymään ja tällöin kaikki datan käsittely loppuu. Yleensä lokidataa kannattaa säilyttää tietoturvamielessä mahdollisimman pitkään, mutta opinnäytetyössä on käytössä varsin rajallinen levymäärä, niin dataa säilötään vain 60 päivän ajan. Ylikirjoitus kooksi määritellään 25Gt. Nämä asetukset on mahdollista tehdä vain API kutsulla ja tässä tapauksessa kutsu toteutettiin palvelimelta itseltään CURL sovellusta hyödyntäen kuvion 8 mukaisesti.

```
curl -X PUT "turpa.hevoi.men:9200/_ilm/policy/retention_policy?pretty" -H 'Content-Type:
application/json' -d'
{
  "policy": {
    "phases": {
      "hot": {
        "actions": {
          "rollover": {
            "max_size": "15GB"
          }
        }
      },
      "delete": {
        "min_age": "60d",
        "actions": {
          "delete": {}
        }
      }
    }
  }
}
'
```

Kuvio 8: Elasticsearch datan siivousasetukset.

Indeksit täytyy määritellä käyttämänä tätä politiikkaa. Helpoin tapa tehdä tämä on määritellä, että kaikki indeksit käyttävät samaa politiikkaa. Tällöin määritellään indeksi kuvioksi *. Lisäksi asetuksiin määritellään, että käytettävä putsauspolitiikka on nimeltään retention_policy ja ylikirjoitus -merkintä on nimeltään rollover-alias. Kuviossa 9 näytetään CURL pyyntö, jolla määritellyt asetukset on asetettu voimaan.

```
curl -X PUT "turpa.hevoi.men:9200/_template/my_template?pretty" -H 'Content-Type:
application/json' -d'
{
 "index_patterns": ["*"],
 "settings": {
  "number_of_shards": 1,
  "number_of_replicas": 1,
  "index.lifecycle.name": "retention_policy",
  "index.lifecycle.rollover_alias": "rollover-alias"
 }
}
'
```

Kuvio 9: Siivouspolitiikan käyttöönotto kaikkiin indekseihin.

5.5.3 Kibana asetukset

Kibanan tarkoitus ELK kokonaisuudessa on tarjota graafinen käyttöliittymä ja se toimii web -pohjaisesti. Datan käsittely tapahtuu muilla palvelimilla, joten Kibana palvelin on hyvin kevyt. Konfigurointeihin ei jouduta tekemään suuria muutoksia, mutta oletusasetuksia joudutaan hieman muuttamaan.

Kibanan asetukset sijaitsevat /etc/kibana/kibana.yml tiedostossa. Tämän sijainti riippuu, kuten myös Elasticsearchin kanssa, siitä, että mikä käyttöjärjestelmä on käytössä kuin myös asennustavasta. Kibana asennettiin vastaavalla tavalla kuin Elasticsearch.

Muutokset kibana.yml konfiguraatiotiedostoon:

```
Server.port: 5601
Server.host: ”palvelimen kokonimi”
Server.name: “palvelimen nimi”
Elasticsearch.hosts: [“http://elasticsearch_palvelimen_nimi:9200”]
Server.ssl.enabled: false
```

Asetukset ovat melko yksiselitteisiä. Elasticsearch.hosts asetus määritelee Elasticsearch palvelimen johon Kibana yhdistetään, server.port käytetyn portin, server.host määrittää palvelimen koko nimen ja server.name lyhyen nimen. Lisäksi server.ssl.enabled arvo määrittelee, että käytetäänkö palvelimen kanssa SSL suojausta.

5.5.4 Filebeat asetukset

Filebeat on ainut Beat -sovelluksista, joka asennetaan erilliselle palvelimelle. Muut opinnäytetyössä käytetyt Beatit asennetaan suoraan laitteisiin, joista data ohjataan eteenpäin käsiteltäväksi. Filebeat toimii keräilypalvelimella, eli useat laitteet ja palvelut lähettävät lokinsa Filebeat palvelimelle, joka toimittaa ne jatkokäsittelyyn Logstashiin.

Filebeat asetuksia joudutaan tekemään useita. Jokaiselle integraatiolle, jotka hyödyntävät Filebeatia joudutaan mahdollisesti tekemään omansa, ellei tiedostomuodot ole identtisiä. Nämä asetukset käydään läpi opinnäytetyössä myöhemmin jokaisen integraation kanssa erikseen. Filebeatin peruskonfiguraatio on muutettu niin, että se hakee konfigurointinsa erillisistä tiedostoista, jotka sijaitsevat asennuskansiossa config.d hakemistossa. Ilman tätä konfiguraatiota kaikkien lähteiden konfiguraatiot olisivat yhdessä tiedostossa, mikä tekisi hallinnasta sekavaa. Kuvio 10 kuvaa Logstashin yleistä konfiguraatiota, jossa on määritelty tiedostopolku, josta konfiguraatiota haetaan, sekä mihin lokit lähetetään käsittelyn jälkeen.

```
#---------------------------- Input configuration --------------------------------
filebeat.config.inputs:
  enabled: true
  path: inputs.d/*.yml
  reload.enabled: true
  reload.period: 10s
#---------------------------- Logstash output --------------------------------
output.logstash:

  # The Logstash hosts
 hosts: ["192.168.2.216:5044"]
 ssl.enabled: false
```

Kuvio 10: Logstash yleinen konfiguraatio.

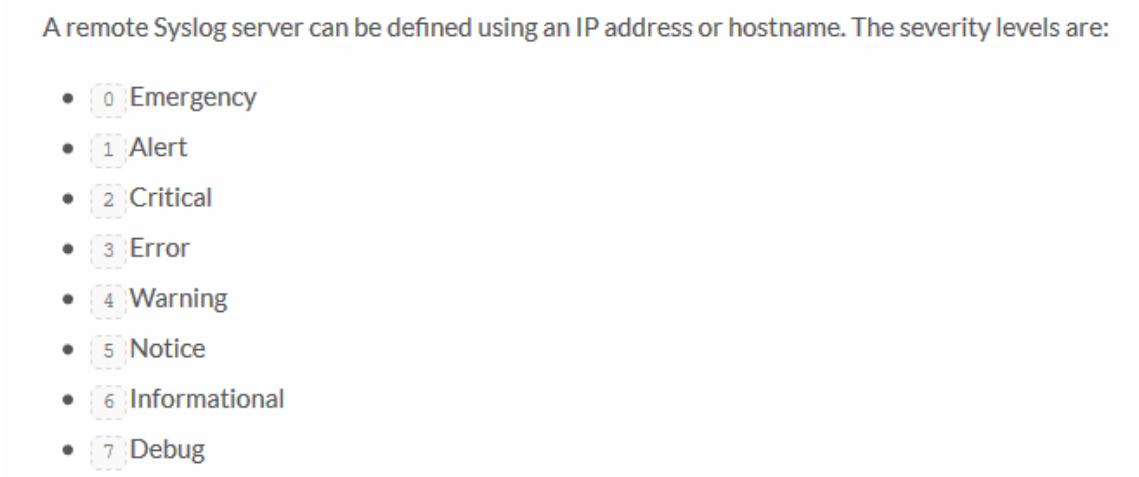
5.6 Lokilähteiden läpikäynti ja integrointi järjestelmään

Lokilähteet tulee integroida ELK järjestelmään käyttäen hyväksi eri tarkoitukseen soveltuvaa Beats sovellusta. Lisäksi Logstashiin tulee tehdä soveltuvat konfiguroinnit, joilla lokidata käsitellään oikein. Tämä voi vaatia hyvinkin paljon konfigurointia, koska lokilähteet eivät ole määrämuotoisia vaan jokaiselle integroitavalle kohteelle tulee luoda omat asetuksensa. Tietyt osiot voidaan aina kopioida, mutta pääosin kaikki joudutaan tehdä erikseen. Mikäli data ei vaadi erillistä käsittelyä vaan on hyvin määrämuotoista, voidaan se lähettää suoraan Elasticsearch tietokantaan ilman, että data toimitetaan ensin Logstashiin.

Jokaisen integroitavan lähteen kohdalta käydään läpi lyhyesti syy lokilähteen integrointiin, sekä mahdolliset konfiguraatiot. Lokilähteiden integroinnin jälkeen data ei ole vielä kovin käyttökelpoista, mutta se on saatavilla myös Kibanassa. Kibanassa tulee tehdä omat visualisointinsa ja hakuehtonsa, että data saadaan näytettyä järkevässä muodossa. Tämä käsitellään myöhemmin opinnäytetyössä.

5.6.1 Reitittimen syslog

Reitittimen lokitiedot kerätään ja lähetetään ELK järjestelmään. Näistä lokitiedoista valvotaan tapahtumia, joita reitittimellä on tapahtunut. Reititin tukee suoraan syslog -lokien lähettämistä ulkoiselle palvelimelle. Asetukset ovat hyvin helppo ymmärtää ja niitä on vain kaksi, lokin kriittisyys ja palvelimen osoite. Kriittisyys määrittelee minkä tasoiset lokit lähetetään eteenpäin ja nämä ovat esitelty Kuviossa 10. Kriittisyysaste asetetaan arvoon 6, koska reitittimestä halutaan melko paljon tietoa, mutta kriittisyys 7 tuottaa liian paljon dataa.



Kuvio 11: Ubiquiti syslog kriittisyysasetikko. (Ubiquiti.)

Syslog tuodaan sisään hyödyntäen Filebeat palvelinta, jolle konfiguroidaan soveltuvat Syslog kuuntelija. Tämä kuuntelee UDP portissa 514, jota Ubiquitin reititin osaa hyödyntää suoraan. Filebeat konfiguraatio on kuvattu kuviossa 12.

```
- type: syslog
  protocol.udp:
      host: "192.168.2.218:514"
  fields_under_root: true
  fields:
   type: syslog
  enabled: true
```

Kuvio 12: Filebeat konfiguraatio.

Filebeat lähettää käsitellyn lokin eteenpäin Logstash palvelimelle, joka käsittelee lokitiedot ja rikastaa lokia, jonka jälkeen data lähetetään Elasticsearch tietokantaan, josta sitä voidaan käsitellä Kibanalla. Kibanalle luodaan soveltuvat näkymät, että lokista saadaan lukukelpoista. Logstash konfiguraatio on kuvattu kuviossa 13.

```
input {
 if [type] == "syslog" {
   beats {
    host => ["192.168.2.216"]
    port => 5044
    ssl => false
    ssl_verify_mode => none
  }
 }
}
output {
 if [type] == "syslog" {
  elasticsearch {
   hosts => "elasticsearch_palvelin:9200"
   index => "logstash-syslog-%{+YYYY.MM.dd}"
   document_type => "%{[@metadata][type]}"
  }
 }
}
```

Kuvio 13: Logstash konfiguraatio reitittimen Syslogille.

Syslog data on nyt integroituna loppuun asti. Data viedään tietokantaan, jossa dataa vasten voidaan tehdä hakuja ja siitä voidaan luoda visualisointeja. Nämä luodaan myöhemmässä vaiheessa opinnäytetyötä.

5.6.2 Verkkohälytystiedot

Verkkohälytystiedot tullaan tuottamaan ilmaisella Security Onion tuotteella. Security Onion tulee tuottamaan hälytyksiä perustuen verkkodataan, jota syötetään toiselta verkossa olevalta kytkimeltä laitteelle. Security Onion toimii sensorina, eli se analysoi liikennettä ja lähettää hälytykset eteenpäin ELK järjestelmään.

Security Onion käyttää taustalla itse myös ELK järjestelmää, eli tässä integraatiossa käytetään valmiita konfigurointeja, jotka ovat jo luotuna Security Onioniin. Tällöin muokattiin valmiita Logstash konfiguraatioita. Konfiguraatioihin vaihdetaan paikallisen Elasticsearch määrityksen tilalle keskitetty Elasticsearch -palvelin, jonka jälkeen data lähetettiin oikealle palvelimelle. Integraatio on valmis ja dataa virtaa tietokantaan ja seuraava vaihe onkin luoda soveltuvat haut ja näkymät Kibanaan.

5.6.3 Verkkovalvonta ja analysointi

Verkkovalvontaa varten asennetaan erillinen järjestelmä, joka käyttää lähteenä samaa dataa kytkimeltä, jota valvotaan Security Onionilla. Tämän järjestelmän ero on se, että se kerää kaiken raakadatan verkosta. Tätä dataa vasten voidaan tehdä analyysiä verkkoliikenteestä

käyttäen järjestelmän omaa käyttöliittymää ja sitä varten voidaan tehdä analyysiä. Tämä tuote on nimeltään Moloch.

Tuote tallentaa datansa oletuksena Elasticsearch tietokantaan ja tämän opinnäytetyön yhteydessä tullaan käyttämään samaa Elasticsearch tietokantaa, johon tallennetaan muutkin tiedot (Moloch). Tuotteen luonteen takia tallennettava datamäärä on erittäin suuri, joten lokin säilömisaika pidetään lyhyenä. Aikaisemmin suoritetun testin perusteella kahden viikon data nykyisestä verkosta vaatii noin 200Gt levytilaa, joten tämä on täysin maksimi, jota voidaan säilyttää. Tämän datan käyttö tapahtuu erillisellä konsolilla, joten sitä varten ei tarvitse tehdä erillistä näkymää Kibanaan.

5.6.4 Päätelaitevalvonta

Päätelaitevalvontaa varten käytetään tuotetta nimeltä Wazuh. Wazuh on ilmaisen lähdekoodin sovellus, jolla voidaan nauhoittaa päätelaitteiden tapahtumia. Lisäksi Wazuhissa on mahdollisuuden suorittaa toimenpiteitä päätelaitteissa. Tuote on hyvin kattava ja se tarjoaa useita erilaisia toimia, kuten haittaohjelmien havainnointia. Tuote ei toimi kuten Antivirus sovellukset, vaan perustuu dataan, jota kerätään ja anomalioihin, joita datassa esiintyy. (Wazuh.)

Tuote toimii päätelaite-palvelin periaatteella, eli tarvitsee oman palvelimensa. Security Onionissa on mukana Wazuh-palvelin, joten erillistä palvelinta ei asenneta. Päätelaitesovellus asennetaan palvelimiin ja kotiverkossa olevaan Windows työasemaan. Palvelinta ei julkaista internetiin, eli laitteet voivat lähettää tietojaan vain, jos ovat kotiverkossa.

Agentit asennetaan laitteille pääasiassa käyttäen Ansiblea, koska asennettavia päätelaitteita on useita. Varsinainen Ansiblen konfiguraatio ei kuulu tähän opinnäytetyöhön, joten asennusta ei kuvata tarkemmin. Data kuitenkin ohjataan konfiguraation mukaisesti Security Onion palvelimelle, josta data ohjataan eteenpäin keskitettyyn Elasticsearch tietokantaan.

5.6.5 Palvelinten pääsynhallinnan lokit

Palvelinten pääsynhallinnan lokit tallennetaan Elasticsearch kantaan. Opinnäytetyössä ei ole käytössä keskitettyä aktiivihakemistoa, kuten Active Directorya, joten palvelimilta tulevat lokit tulee tallentaa suoraan jokaiselta palvelimelta. Tähän voitaisiin käyttää keräilypalvelinta, johon kerätään keskitetysti lokit kaikilta palvelimelta ja lähetetään eteenpäin, mutta opinnäytetyössä analysoitava ympäristö on niin pieni, että keräilyagentti asennetaan kaikkiin palvelimiin erikseen.

Pääsynhallinnan lokien keräämiseen käytetään Auditbeatia, joka on Elasticin tekemä keräin. Asennukseen käytetään Ansiblea ja soveltuvaa työkirjaa, mutta tätä ei käydä tässä opinnäytetyössä enempää läpi. Lokit lähetetään suoraan Elasticsearch kantaan eikä Logstashiin, koska

Kibanassa on valmiiksi luotu näkymä, jota voi käyttää tietojen analysoimiseen. Kibanan näkymä vaatii sen, että data lähetetään sellaisenaan suoraan Elasticsearchiin, niin että sitä ei muuteta matkalla. Tämän lisäksi voidaan myös tarpeen ollen rakentaa omia näkymiä.

Pääsynhallinnan lokien keräämistä varteen tehtiin konfiguraatio, jossa määritellään mitä kerätään ja mihin ne lähetetään. Tämä konfiguraatio löytyy Kuviosta 14. Konfiguraatio on suhteellisen yksinkertainen. Ensin määritellään mitä moduuleita käytetään, joista file_integrity on kommentoitu pois, koska se tuotti liian paljon tapahtumia. Tämän jälkeen määritellään output.elasticsearch kohdassa Elasticsearch tietokanta johon data lähetetään, sekä output.kibana kohdassa Kibana palvelin johon konfiguroidaan näkymät.

```
auditbeat.modules:
- module: auditd
  audit_rule_files: [ '${path.config}/audit.rules.d/*.conf' ]
  audit_rules: |

- module: system
  datasets:
    - host
    - login
    - package
    - process
    - socket
    - user
  period: 10s
  state.period: 12h
  socket.include_localhost: false
  user.detect_password_changes: true

output.elasticsearch:
  hosts: ["elasticsearch_palvelin:9200"]
  index: "auditbeat-%{[agent.version]}-%{+yyyy.MM.dd}"
setup.kibana:
  host: "kibana_palvelin:5601"
```

Kuvio 14: Auditbeat asetukset.

5.6.6 Palvelinten suorituskykytiedot

Palvelinten suorituskykytiedot voivat paljastaa tietoturvapoikkeamia ympäristöissä, vaikkakaan niitä ei pääasiallisesti kerätä tietoturvamielessä. Usein haittaohjelmat tai haitalliset tapahtumat aiheuttavat laitteissa ylimääräistä kuormaa joko suorittimelle tai muistille. Tällöin normaalista poikkeavista kuormista voidaan havaita, että laitteilla on käynnissä tietomurto. Tämän takia, myös tässä opinnäytetyössä kerätään suorituskykydataa palvelimilta. Dataa ei kerätä kuitenkaan muilta päätelaitteilta.

Suorituskykydataa varten palvelimille asennetaan Metricbeat joka on Elasticin rakentama keräin joka voidaan asentaa suoraan laitteille. Metricbeatilla data kerätään suoraan Elasticsearch tietokantaan käyttämättä Logstashia, koska tällöin voidaan käyttää valmiita Kibanan näkymiä tiedon analysoimiseen. Kuten auditbeatinkin kanssa voidaan myös Metricbeatin kanssa luoda omat näkymänsä, mikäli Elasticin valmiiksi luomat näkymät eivät ole riittävät.

Metricbeatin konfigurointi on todella yksinkertaista. Konfiguraatiossa määritellään moduulit, joista suorituskykyä mitataan, tämän opinnäytetyön puitteissa mitataan vain käyttöjärjestelmätason tietoa. Lisäksi konfiguroinneissa määritellään mihin tieto lähetetään ja mitä Kibana-palvelinta datan analysoimiseen käytetään. Opinnäyteyössä käytettävä konfiguraatio löytyy kuviosta 15.

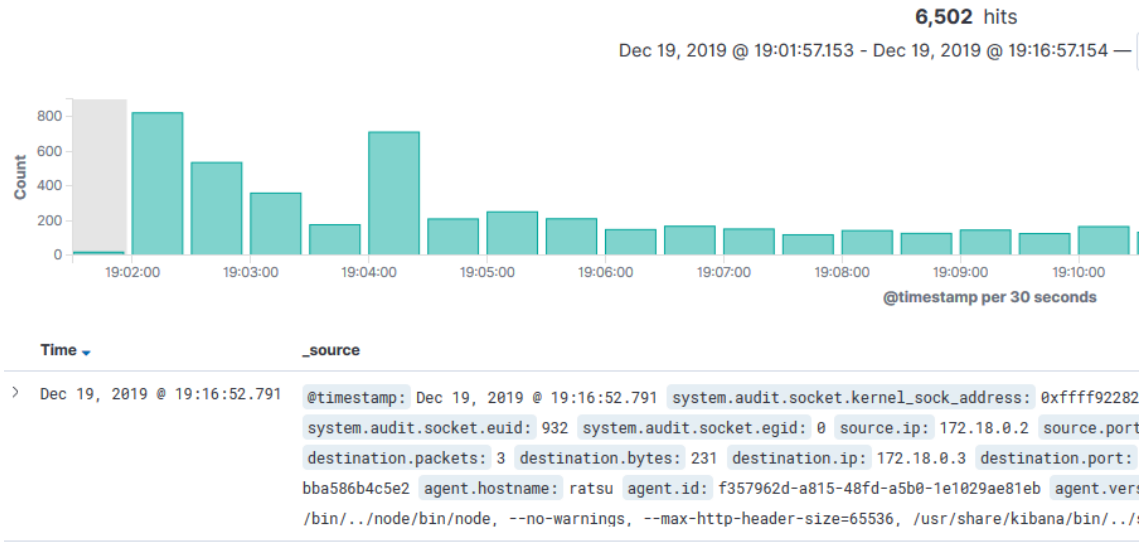
```
metricbeat.modules:
- module: system
  metricsets:
["cpu","memory","network","diskio","filesystem","uptime","core","entropy",fsstat","load","process
","process_summary","raid","socket","socket_summary"]
  enabled: true
  period: 15s
  processes: ['.*']

output.elasticsearch:
  hosts: ["elasticsearch_palvelin:9200"]
setup.kibana:
  host: "kibana_palvelin:5601"
```

Kuvio 15: Metricbeatin konfiguraatio.

5.7 Tallennettujen hakujen luonti lokilähteiden tarpeisiin

Lokidata on viety Elasticsearch tietokantaan ja Kibanaan on luotu tarvittavat indeksit, joita vastaan voidaan luoda tallennettuja hakuja ja visualisointeja. Tallennetut haut voivat sisältää valmiiksi tehtyjä suodattimia, sekä erilaisia muokkauksia, joita on tehty oletuksena näkyvään lokinäkymään.

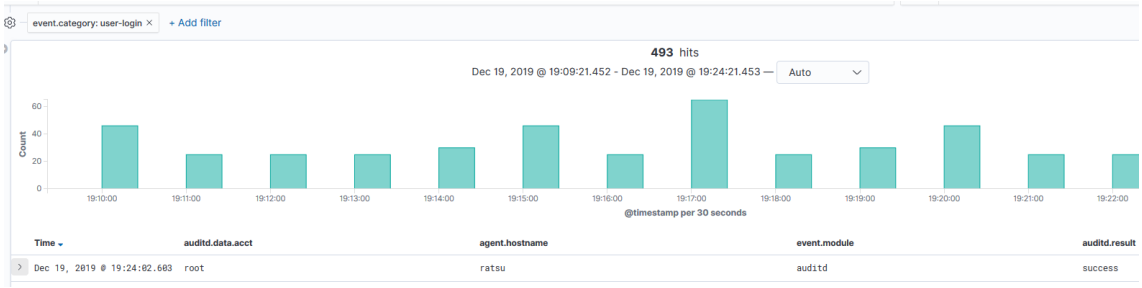
Sovelluksen oletusnäkymä näyttää kenttinä pelkästään tapahtuman ajan ja tämän lisäksi koko sisällön tapahtumasta. Tämä tekee näkymästä lähes käyttökelvottoman, mikäli näkymää ei muokata omiin tarpeisiin soveltuvaksi. Lisäksi oletuksena näkymässä näkyy kaikki tapahtumat ilman suodatusta, joten datan läpikäynti voi olla hyvin haastavaa. Kuvio 16 esittää esimerkkikuvan oletusnäkymästä ilman mitään muokkauksia.



Kuvio 16: Kibana hakujen oletusnäkymä.

Tässä näkymässä näytetään aikaisemmin luotu pääsynhallinta-loki. Tässä lokissa on monia erityyppisiä tapahtumia, mutta esimerkkinä tästä voidaan luoda näkymä, jossa näkyy vain kirjautumistapahtumat. Tästä näkymästä on helppo nähdä kaikki tapahtuvat kirjautumiset laitteilta, joita valvotaan. Dataa on helppo suodattaa lisää, jolloin datana analysoiminen helpottuu lisää.

Suodattimia voidaan tehdä joko käsin, tai sitten viemällä kursori näkymässä olevan datan päälle, josta voidaan helposti suodattaa kaikki vastaava data pois kokonaan, tai vaihtoehtoisesti näyttää pelkästään tähän osuva data. Kuvio 17 kuvastaa näkymää, jossa pääsynhallinta-loki on suodatettu vain kirjautumistapahtumiin ja näkyviin on määritelty sarakkeet Käyttäjätili, laite, johon kirjauduttiin, tapahtuma ja lopputulos (onnistui/epäonnistui).



Kuvio 17: Sisäänkirjautumisnäkymä.

Toinen hyvä esimerkki tallennetusta hausta on haku, jossa on nähtävillä DNS kutsut. Tätä varten joudutaan tekemään suodatin, joka suodattaa verkkodatasta pelkästään DNS kyselyt. Tällainen voidaan tehdä useilla eri tavoilla, mutta yksi yksinkertainen tapa on määitellä event_type kenttä. Tämän jälkeen lokista tarkastetaan tähän näkymään soveltuvat kentät,

jotka ovat kellonaika, lähde IP-osoite, kyselyn tyyppi, itse kysely, sekä DNS palvelimen vastaus. Tällöin näytettävä data on helposti luettavissa ja sitä vasten voidaan tehdä helposti hakuja tai lisäsuodatusta. Tämä tallennettu haku on esitetty kuviossa 18.

	Time	source_ip	query_type_name	query	answers
›	Dec 20, 2019 @ 17:40:51.731	192.168.2.80	A	lh5.googleusercontent.com	googlehosted.l.goog
›	Dec 20, 2019 @ 17:40:46.149	192.168.2.80	AAAA	tools.google.com	tools.l.google.com,
›	Dec 20, 2019 @ 17:40:46.149	192.168.2.80	A	tools.google.com	tools.l.google.com,

Kuvio 18: Tallennettu haku DNS kyselyjen tarkastamiseen.

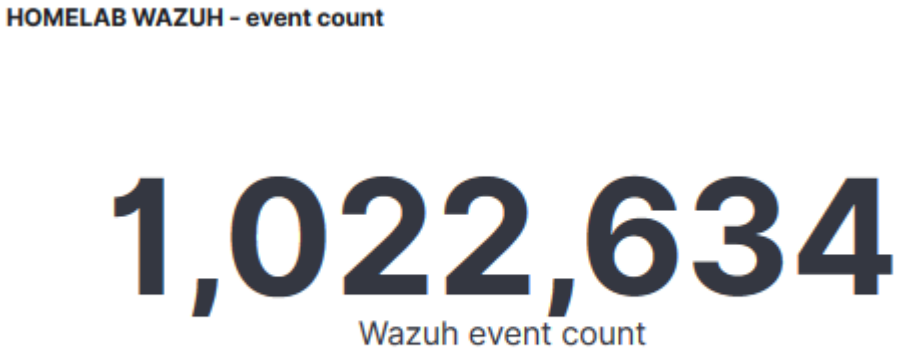
Opinnäyteyössä tarvitaan useita erilaisia tallennettuja hakuja, mutta ei ole tarkoituksenmukaista, että jokainen tallennettu haku esitellään. Täten, nämä kaksi tallennettua hakua toimii esimerkkinä ja tarkoituksena on antaa kuva, miten paljon tallennetut haut voivat selkeyttää datan selaamista.

Opinnäytetyötä varten luotiin 9 erilaista tallennettua hakua.

5.8 Visualisointien luonti

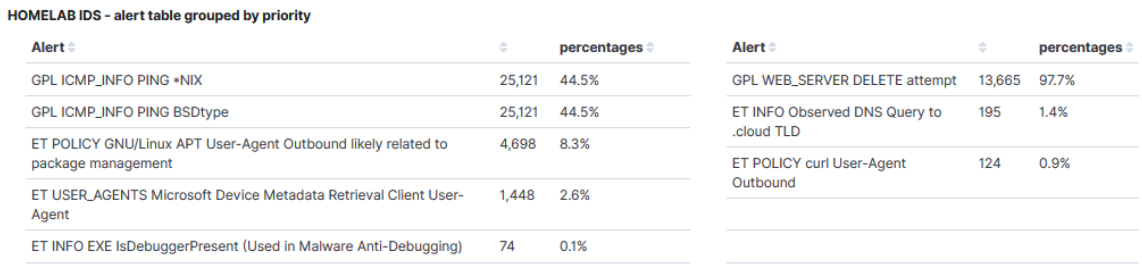
Kibanalla voidaan luoda useita erilaisia visualisointityylejä. Tyyleinä voidaan käyttää esimerkiksi, piirakkakaaviota, karttakaaviota, taulukoita, pylväsdiagrammia ja lämpökarttaa. Y-akselille voidaan määritellä tilastollisia arvoja, kuten keskiarvo, summa, minimi, maksimi, lukumäärä, sekä yksilöllisten esiintymien lukumäärä. X-akselille voidaan määritellä aika-alue, päivämäärä, termi (tarkoittaa kenttää datan sisällä), suodatin ja merkittävä termi. Lisäksi X-akseli voidaan jakaa myös alikategorioihin. (Rovnik.)

Perimmäinen tarkoitus visualisoinneilla on luoda helposti ymmärrettäviä otantoja suuresta määrästä dataa. Visualisointeja voidaan käyttää eri näkökulmista ja usein niitä käytetään esittämään liiketoiminnan kannalta merkittäviä lukuja. Tässä opinnäytetyössä visualisointeja luodaan lokidatasta tietoturvanäkökulmasta, lisäksi luodaan visualisointeja, jotka havainnoivat ympäristöön liittyvää statistiikkaa. Kuvio 19 sisältää visualisoinnin päätelaitevalvonnan tapahtumien kokonaismäärästä.



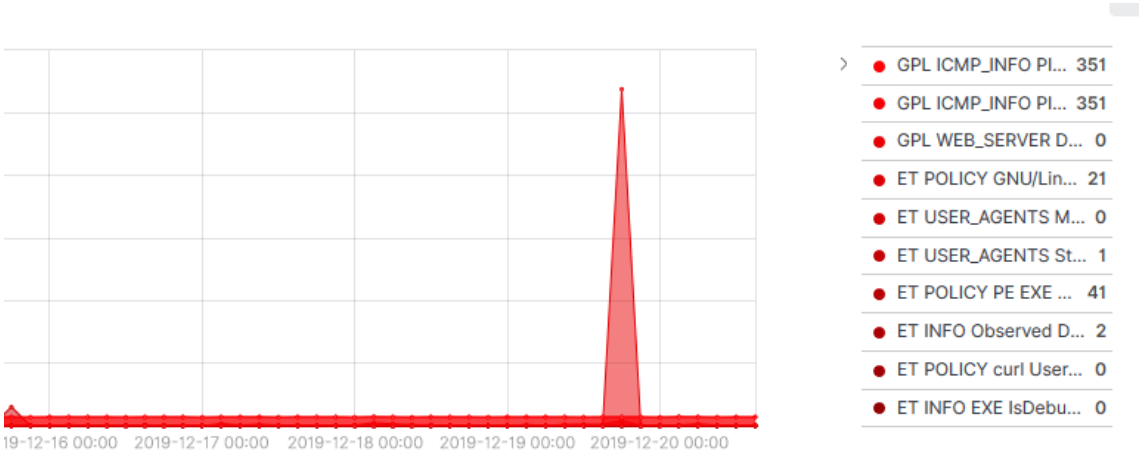
Kuvio 19: Päätelaitevalvonnan tapahtumien määrä.

Tietoturvamielessä huomattavasti tärkeämpi visualisointi on taulukko, jossa on listattuna verkkotapahtumia valvovan sensorin hälytykset. Taulukossa on kolme eri saraketta, joista jokainen kuvaa tapahtumien vakavuutta. Mitä pienempi numero on, sitä vakavampi tapahtuma on ja sitä suuremmalla syyllä sitä tulisi tutkia lisää. Tämä visualisointi luodaan useampien kenttien pohjalta, hyödyntäen sekä hälytyksen nimeä, että sensorin antamaa prioriteettiarvoa. Tämä visualisointi on esitetty kuviossa 20. Kuviota on hieman pienenennetty, koska muuten kuvasta ei saisi selvää.



Kuvio 20: Verkkosensorin hälytystiedoista luotu visualisointi.

Visualisointeja tulee olemaan erittäin suuri määrä, vähintäänkin kymmeniä, joten opinnäytetyössä ei esitellä näitä kaikkia. Kuitenkin aikaisemmat visualisoinnit on perustunut täysin esitettyyn dataan. Kibana pystyy esittämään visualisointeja myös graafisesti, joka tekee datasta vielä paremmin luettavaa tietyissä tapauksissa. Kuviossa 21 esitellään visualisointia, jossa nähdään aikajana verkkosensorin tuottamista hälytyksistä. Aikajanassa esiintyvät poikkeamat normaaliin ovat usein kiinnostavia ja näitä täytyy tutkia lisää.



Kuvio 21: Verkkosensorin hälytysaikajana.

Tätä opinnäytetyötä varten luotiin yhteensä 38 erilaista visualisointia. Opinnäytetyössä käytettiin kuutta erilaista tuotteen valmistajan valmiiksi luomaa visualisointia.

5.9 Yhtenäisen käyttöliittymän luominen (”Dashboard”)

Kibanassa voidaan luoda käyttöliittymiä, jotka tunnetaan nimellä Dashboard. Nämä Dashboardit ovat kokoelma erilaisia visualisointeja, sekä tallennettuja hakuja, jotka muodostavat loogisen kokonaisuuden. Dashboardien tarkoitus on esittää data helposti luettavassa muodossa, josta voidaan vetää johtopäätöksiä yhdellä vilkaisulla, sekä mahdollistaa datan tarkemman tarkastelun helpolla suodattimien lisäyksellä. (Elastic.)

Lähtökohtaisesti Kibanassa ei ole Dashboardeja laisinkaan, kuten ei ole myöskään tallennettuja hakuja tai visualisointeja. Kibanan logiikan mukaisesti kannattaa tehdä ensin halutut visualisoinnit ja tallennetut haut, jonka jälkeen niistä voidaan rakentaa kokonaisuus, eli Dashboard. Elastic on luonut myös valmiista Dashboard näkymiä, jotka istuvat hyvin joihinkin kokonaisuuksiin, kuten esimerkiksi yksittäisen palvelimen suorituskyvyn tutkimiseen. Suurin osa opinnäytetyössä tehdyistä Dashboardeista, visualisoinneista ja tallennetuista hauista on kuitenkin tehty täysin itse, koska valmiit näkymät eivät sovellu hirveän hyvin opinnäytetyössä käytettäväksi.

Lokilähteitä, joita halutaan analysoida, on useita, joten myös Dashboardeja, joita opinnäytetyössä luotiin on enemmän kuin yksi. Kaiken datan sisältäminen yksittäiseen Dashboardiin tekisi näkymästä äärimmäisen sekavan, joten usean Dashboardin ratkaisu on oikea opinnäytetyön kannalta. Yksi visualisoinneista, jota Kibana tarjoaa, on nimeltään Markdown ja tällä voidaan luoda yksinkertainen navigointimenu, jolla voidaan siirtyä näkymästä toiseen.

Opinnäyteyötä varten luotiin 7 erilaista Dashboard näkymää. Näillä näkymillä on opinnäytetyön kantilta hyvin selkeät tarkoitukset:

- Home – kotinäkymä, joka palautetaan oletuksena. Tässä näkymässä käsitellään yleisiä numeerisia arvoja, sekä näytetään kooste hälytyksistä, joita kotiverkossa on nähtävillä. Kotinäkymä on esillä Kuviossa 22.

- Performance – suorituskykytiedot, jossa on nähtävillä yleisiä yhteenvetoja kaikista laitteista, jotka tuottavat suorituskykytietoa palveluun.

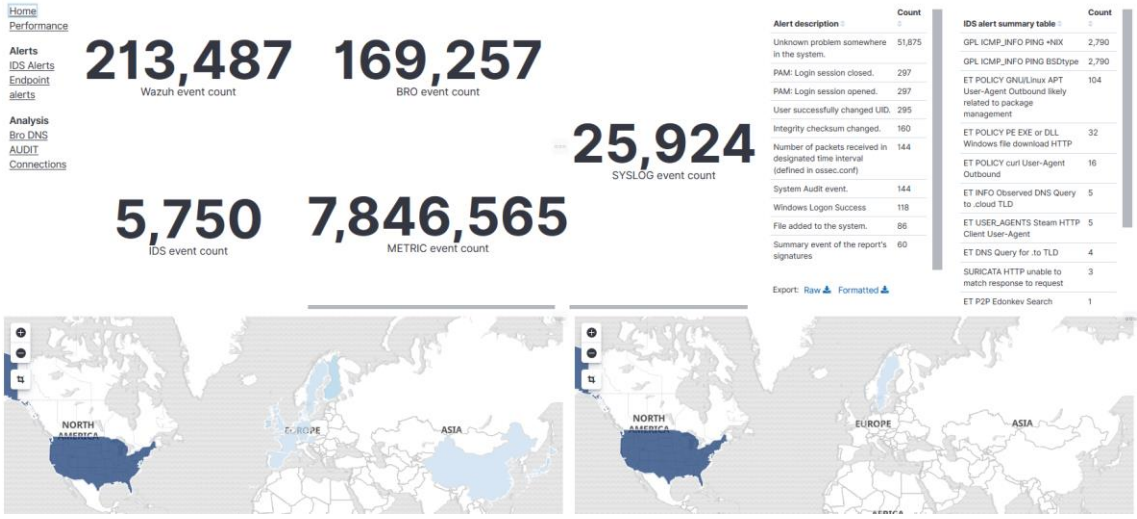
- IDS Alerts – tämä näkymä näyttää yhteenvedon tunkeilijan havaitsemisjärjestelmän tuottamista hälytyksistä.

- Endpoint alerts – näkymässä on nähtävillä päätelaitteista havaitut hälytykset. Tässä näkymässä voidaan myös analysoida päätelaitteiden tapahtumia.

- Bro DNS – näkymä on rajattu näyttämään statistiikkaa DNS kyselyistä, sekä esittämään analysointia varten helposti ymmärrettävä näkymä tallennettuun lokidataan.

- AUDIT- näyttää pääsynhallinnan lokeista koostettuja yhteenvetoja, sekä tarjoaa helposti ymmärrettävän näkymän pääsynhallinnan lokidatan analysointia varten.

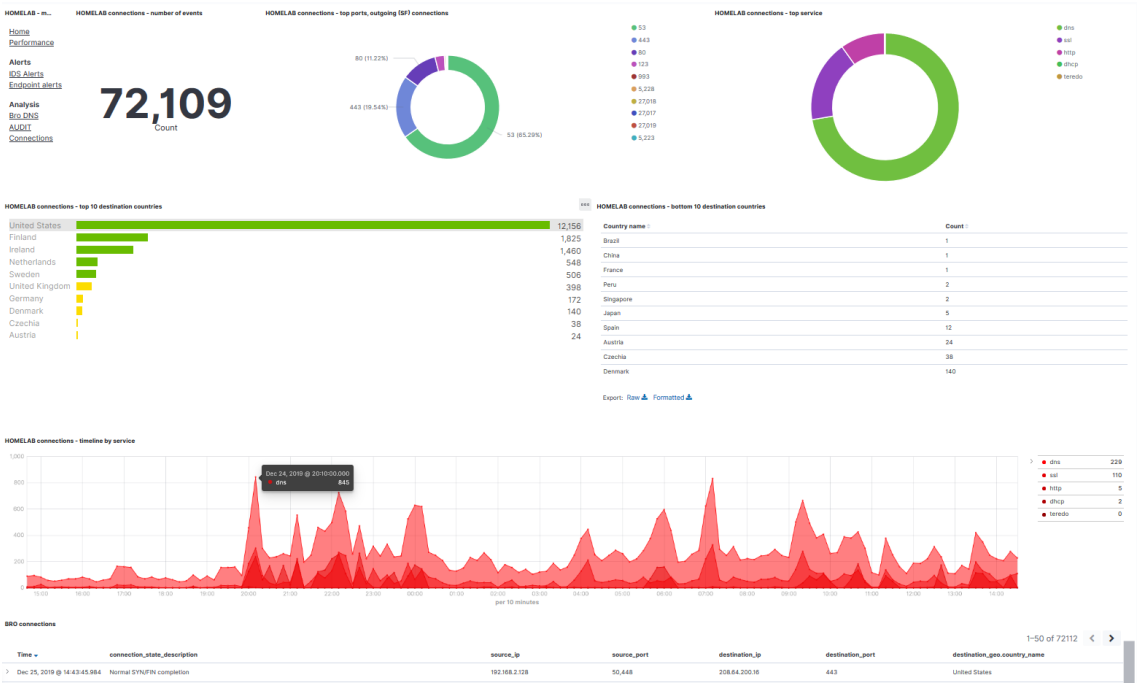
- Connections – koostaa netflow dataa, jota voidaan analysoida näkymässä helposti.



Kuvio 22: Kotinäkymä rakennetusta ympäristöstä.

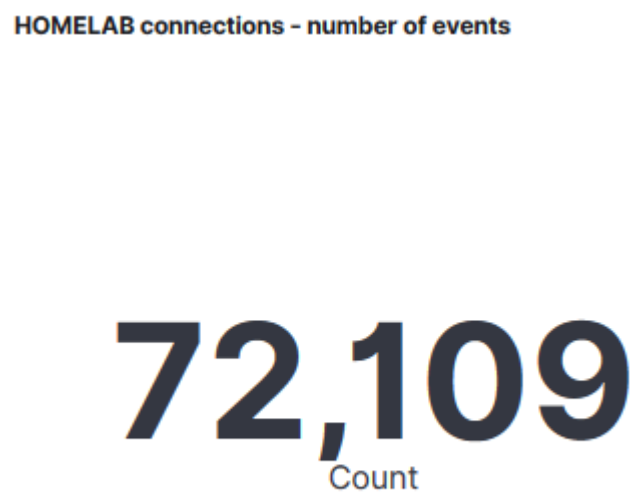
Nämä näkyvät eivät ole helppolukuisia, tai helposti ymmärrettäviä, mikäli aihealue ei ole ennestään tuttu. Tämän takia yksi luoduista näkymistä käsitellään kokonaisuudessaan kuvin, sekä selityksin läpi, jotta tarkoitus näkymille selviäisi paremmin. Kaikkia luotuja näkymiä ei käydä läpi, koska se ei ole tarpeen mukaista opinnäytetyön luonteen kannalta. Helpoiten lähestyttävä dashboard on Connections dashboard joka näyttää statistiikkaa kaikesta verkossa

liikkuvasta verkkodatasta Netflow muodossa. Kuvio 23 näyttää tämän Connections dashboard yleisnäkymän.



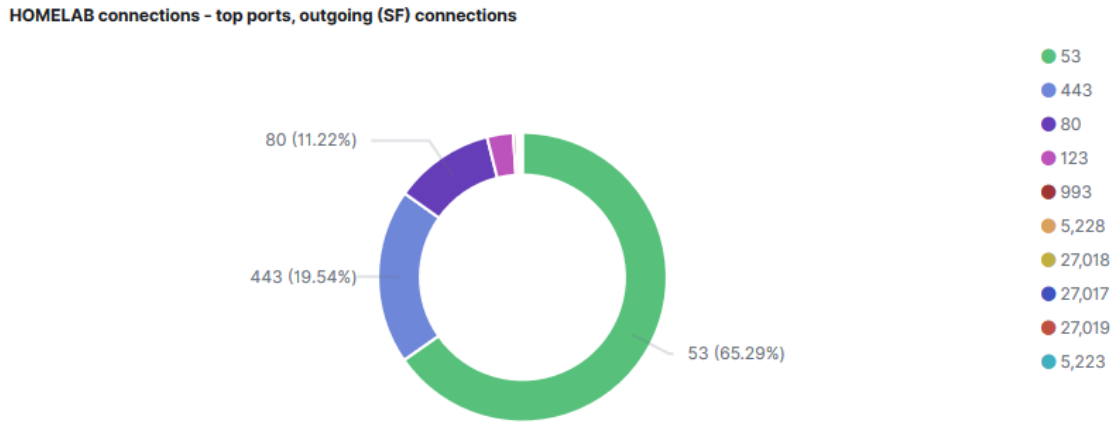
Kuva 23: Opinnäyteyötä varten luotu connections dashboard.

Tässä kyseisessä dashboardissa on luotuna 5 erilaista visualisointia ja yksi tallennettu haku. Ensimmäinen visualisointi näyttää kuinka paljon tapahtumia kyseisessä lokissa on yhteensä analysoidulla ajanjaksolla. Tämä on esillä kuviossa 24.



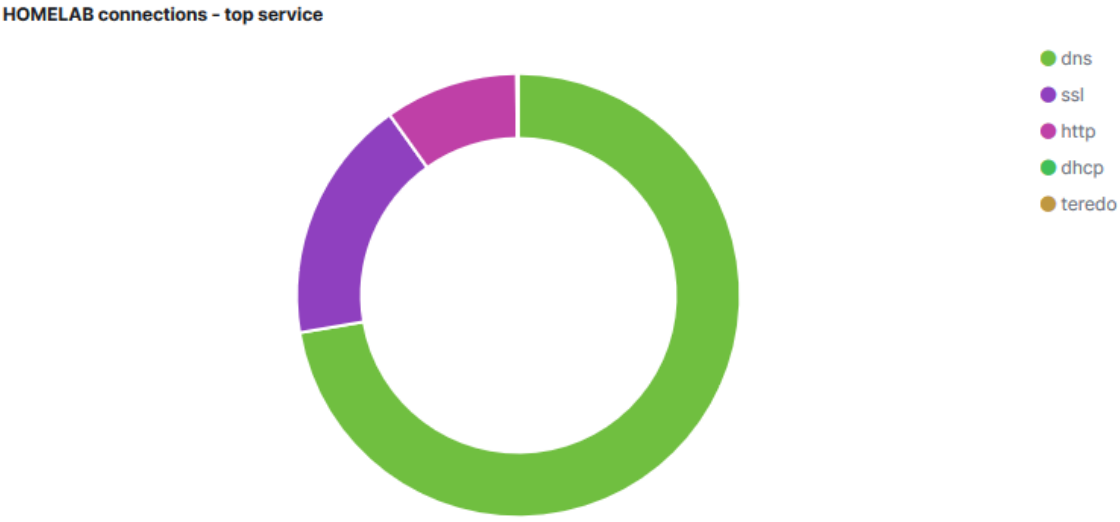
Kuvio 24: Connections dashboardin tapahtumien määrän visualisointi.

Seuraava kuvio näyttää piirakkadiagrammilla 10 yleisintä kohdeporttia johon verkossa on yhdistetty. Visualisointi on rajoitettu ulospäinsuuntautuviin yhteyksiin, koska muuten TCP protokollan mukainen syn-ack malli tuottaisi näkymään suuren määrän korkeita portteja, jotka tekisivät näkymästä vaikean luettavan ja tuottaisi turhaa tietoa. Tämä visualisointi on esillä kuviossa 25.



Kuvio 25: Yleisimmät kohdeportit visualisointi.

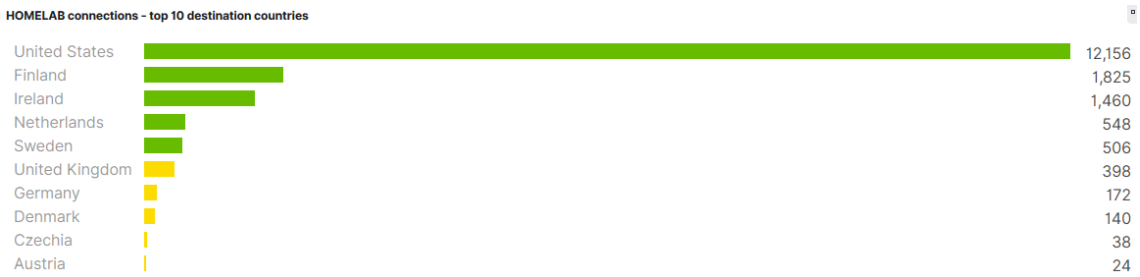
Seuraava visualisointi kuvaa yleisimpiä kohdepalveluja joihin verkossa on yhdistetty analysoitavana ajanjaksona. Visualisointi on kuvattu kuviossa 26.



Kuvio 26: Yleisimpien kohdepalvelujen visualisointi.

Seuraavaksi Dashboardissa näkyy visualisointi, joka näyttää 10 maata, joihin analysoitavana ajanjaksona on yhdistetty kaikista useimmin. Tässä näkymässä on määritelty, että palkki muuttuu keltaiseksi, mikäli yhteyksiä on alle 500 ja punaiseksi mikäli yhteyksiä on alle 20.

Tietoturvamielessä ne maat, joihin yhdistetään useimmin eivät ole kiinnostavia, vaan maat johon yhteyksiä muodostetaan vain hieman. Visualisointi on kuvattu kuviossa 27.



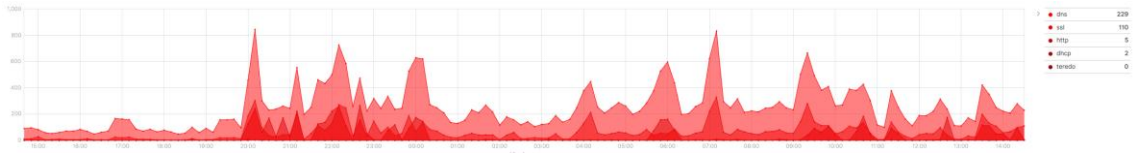
Kuvio 27: Visualisointi, joka näyttää maat joihin yhteyksiä on muodostettu eniten.

Seuraava visualisointi kuvaa maita, joihin on yhdistetty vähiten. Tämä on tietoturvamielessä huomattavan paljon mielenkiintoisempaa. Värikoodeja tässä visualisoinnissa ei ole, koska visualisointi on luotu eri työkalulla. Visualisointi on kuvattu kuviossa 28.

Country name	Count
Brazil	1
China	1
France	1
Peru	2
Singapore	2
Japan	5
Spain	12
Austria	24
Czechia	38
Denmark	140

Kuvio 28: Visualisointi, joka kuvaa maita joihin yhteyksiä on vähiten.

Viimeinen visualisointi, joka dashboardiin on tuotu näyttää aikajanan. Aikajana on määritelty näyttämään oma metriikkansa jokaiselle eri palvelulle. Tämän aikajanan tarkoitus on helpottaa tutkintaa, mikäli on tarvetta tutkia verkkodataa ja kuinka yhteyksiä on muodostettu yksittäiseen osoitteeseen. Lisäksi näkymässä näkyy trendi ja mikäli yhtäkkiä trendi muuttuu suuresti, niin voidaan olettaa, että verkossa on tapahtunut anomalia. Usein tällaiset anomaliat voivat liittyä tietomurtoon. Visualisointi on esillä kuviossa 29, joskin valitettavan pienenä.



Kuvio 29: Visualisointi verkkotapahtumien aikajanasta jaoteltuna palvelulla.

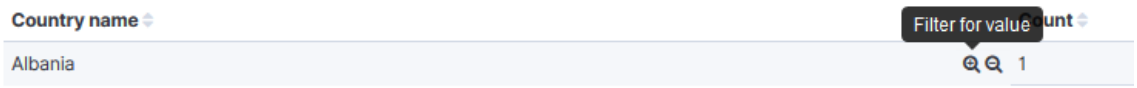
Visualisointien lisäksi dashboardissa on käytetty yhtä tallennettua hakua. Tämän haun tarkoitus on koostaa kaikista käytössä olevista kentistä yhtenäinen ja helposti luettavissa oleva näkymä. Tässä lokissa kenttiä on kymmeniä, mutta nopealle analysoimiselle niistä oleellisia on vain muutama. Nämä oleelliset kentät on lisätty tallennettuun hakuun esille ja jokaisen näkyvän rivin voi tarvittaessa laajentaa, jolloin kyseiseltä riviltä on nähtävissä kaikki muutkin kentät. Tämä tallennettu haku on kuvattu kuviossa 30. Näkymästä on jäänyt yksi sarake pois, joka on kohdemaa.

Time ▾	connection_state_description	source_ip	source_port	destination_ip	destination_port
Dec 25, 2019 @ 14:43:45.984	Normal SYN/FIN completion	192.168.2.128	50,448	208.64.200.16	443
Dec 25, 2019 @ 14:43:45.981	Normal SYN/FIN completion	192.168.2.211	45,347	192.168.2.1	53
Dec 25, 2019 @ 14:43:37.549	Normal SYN/FIN completion	192.168.2.80	44,002	8.8.8.8	53

Kuvio 30: Tallennettu haku, joka kuvaa verkkoyhteyksiä.

Opinnäytetyössä luotuja näkymiä voidaan sellaisenaan hyödyntää jo verkon valvontaan. Näkymät näyttävät helposti hälytysdataa, mikäli sellaisia ympäristössä havaitaan. Tämän lisäksi näkymät mahdollistavat tallennetun datan analysoinnin. Kibanaan luodut dashboardit toimivat niin, että kun näkymään tehdään suodatin, niin kaikki visualisoinnit näyttävät vain suodattimien suodattamaa dataa. Tällöin voidaan tehdä esimerkiksi yksinkertainen suodatin, jossa määritellään, että näytetään vain dataa, jossa kohde IP-osoite on 8.8.8.8. Tällöin kaikki visualisoinnit muuttavat näkymää tämän suodattimen mukaiseksi.

Lisäksi suurin osa visualisoinneista tukee ominaisuutta, jolla suodattimen voi luoda klikkaamalla suoraan visualisoinnista. Aikaisemmin esitellystä visualisoinnista voidaan esimerkiksi suoraan suodattaa kohdemaaksi vain Albania. Tällöin kaikki visualisoinnit muuttuvat, eikä vain se visualisointi, jossa suodatin on tehty. Tämä toiminto on kuvattu kuviossa 31.



Kuvio 31: Datan suodatus visualisoinnin sisältä.

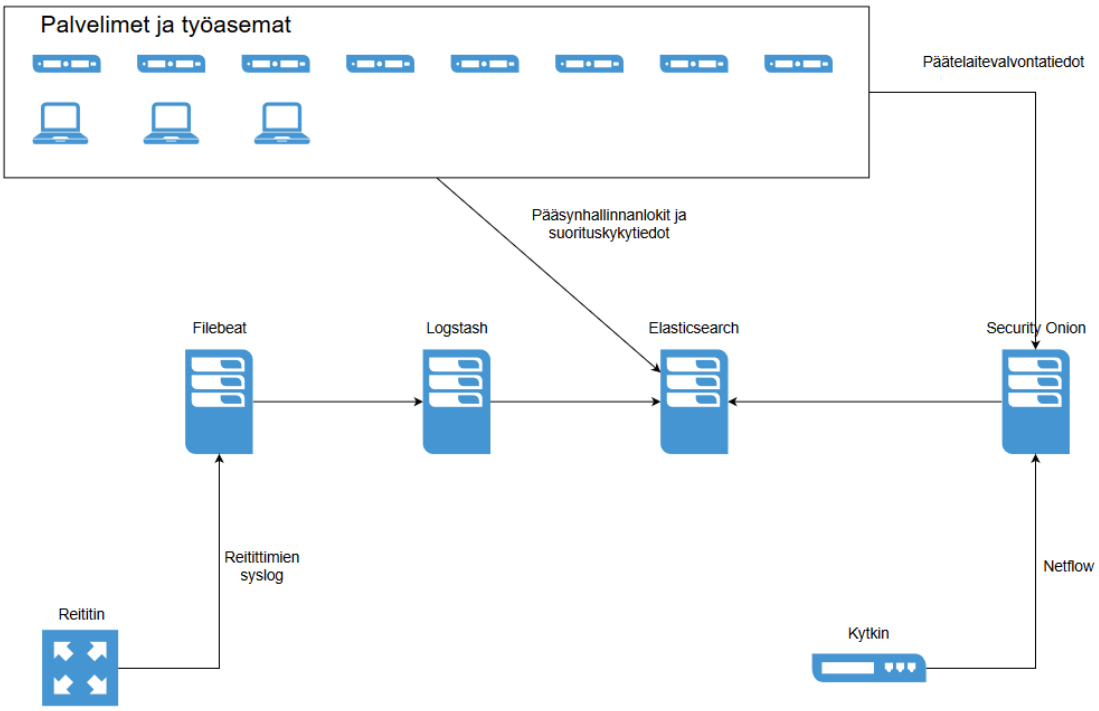
6 Asennetun ELK -järjestelmän kuvaus ja yhteenveto

Asennettu ympäristö koostuu useista valvottavista laitteista ja lokilähteistä. Lokilähteistä kerätty data lähetettiin Filebeat, Logstash ja Elasticsearch palvelimille suoraan riippuen siitä mikä oli lokilähteen kantilta järkevintä. Lokilähteet on kuvattu taulukossa 4.

Lokilähde	Käyttötapa	Mihin data integroidaan
Kytkimen verkkopeilaus.	Verkkohälytystietojen toimittaminen, verkkoliikenteen analysointi.	Security Onioniin, josta se integroidaan Elasticsearch kantaan.
Palvelinten pääsynhallintaloki.	Anomalioiden analysoiminen, kirjautumisen analysointi.	Elasticsearch kantaan suoraan, mahdollistaa valmiiden Kibana visualisointien käytön.
Palvelinten suorituskykytiedot.	Anomalioiden havaitseminen, suorituskykymittaus ja levytilan käytön monitorointi.	Elasticsearch kantaan, joka mahdollistaa valmiiden visualisointien käytön.
Palvelinten ja työasemien tapahtumatiedot.	Hälytysten nostaminen ja päätelaitetapahtumien analysointi.	Security onioniin, josta integrointi Elasticsearch kantaan.
Reitittimen syslog lokit.	Reitittimen tapahtumien analysointi.	Filebeatiin, josta integraatio Logstashiin ja sitä kautta Elasticseaschiin.

Taulukko 4: Lokilähteiden integroinnit.

Lokilähteet tuottivat vaihtelevan hyödyllistä tietoa. Reitittimen syslog -loki oli lokilähteistä hyödyttömin, mutta kytkimen tuottamasta verkkoliikenteen peilatusta datasta saatiin erittäin paljon hyödyllistä lokia, jota voitiin sekä valvoa, että analysoida tarpeen mukaan. Opinnäytetyössä käytettiin Security Onion tuotetta johon verkkodata integroitiin ja sitä myöden data saatiin myös tuotettua asennettuun ELK-järjestelmään. Integraatiot on kuvattu arkkitehtuurikuvana kuviossa 32.



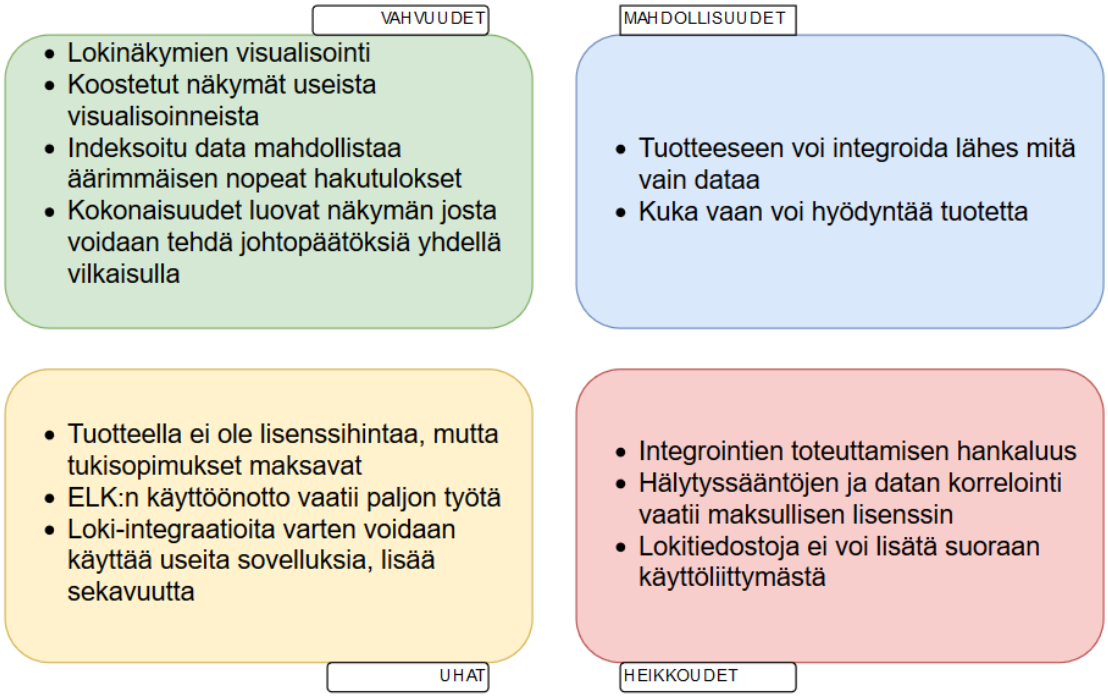
Kuvio 32: Lokilähteiden arkkitehtuuri.

Kuten integraatioiden arkkitehtuurikuva kuvastaa, niin tämän opinnäytetyön kantilta Filebeatin ja Logstashin roolit jäivät hyvin pieneksi, koska suurin osa datasta tuli Security Onionista, joka käsitteli datat sisäisesti käyttäen samoja komponentteja. Tällöin integraatiota ei ollut järkevää tehdä kahta kertaa. Palvelinten ja päätelaitteiden pääsynhallinnan lokit ja suorituskykydata integroitiin käyttäen järjestelmän valmistajan omia työkaluja, jolloin integrointi oli järkevämpää tehdä suoraan Elasticsearchiin, että saatiin valmistajan valmiiksi luomat näkymät käyttöön.

ELK-järjestelmän lokianalysointia varten luotiin useita erilaisia tallennettuja hakuja, visualisointeja ja koottuja näkymiä näistä, jotka mahdollistivat varsinaisen lokianalyysin. Koottuja näkymiä, eli dashboardeja, opinnäytetyössä luotiin yhteensä 7 kappaletta. Visualisointeja luotiin 38, jonka lisäksi käytettiin kuutta erilaista valmiiksi luotua näkymää. Lisäksi tallennettuja hakuja luotiin 9 kappaletta.

7 SWOT-analyysi

ELK:n kyvykkyyksistä lokianalyysin ja valvonnan kannalta tehtiin SWOT-analyysi, joka on kuvattuna kuviossa 33. Analyysissä huomioitiin opinnäytetyön tarkoitus, eli analyysiä mietittiin sen kannalta, että miten lokianalyysiä ja valvontaa voidaan tehdä ja tehostaa ELK tuotteella.



Kuvio 33: SWOT-analyysi.

7.1 SWOT-analyysin vahvuudet

ELK-järjestelmän ensimmäiseksi vahvuudeksi on määritelty lokinäkymien visualisointi. Tuote osoittautui erittäin kyvykkääksi näyttämään lokidatan erilaisissa graafeissa, kuten esimerkiksi piirakkadiagrammissa. Tämä tekee tuotteen näyttämästä datasta helposti ymmärrettävää, kun se ei perustu pelkästään taulukkonäkymään. Toinen vahvuus on näistä kuvatauista visualisoinneista muodostetut koostetut näkymät. Näissä näkymissä voidaan tehdä kokonaisuuksia, joissa voidaan näyttää useita samaan lokilähteeseen viittaavia kokonaisuuksia. Tällöin yhdellä vilkaisulla voidaan saada suuri määrä tietoa helposti käsiteltävässä muodossa.

Kolmas vahvuus on indeksoitu data. Tämä tarkoittaa sitä, että datasta tehdään indeksi, joka nopeuttaa datan läpikäyntiä. Käytännössä tämä tekee hauista äärimmäisen nopeita. Neljäs vahvuus liittyy vahvasti toiseen vahvuuteen, mutta tällä tarkoitetaan, että myös eri tyyppisistä yhdistetyistä visualisointikoosteista voidaan tehdä vielä isompia koostenäkymiä. Lisäksi tuotteessa voidaan tehdä navigointipalkki, jolla voidaan siirtyä näkymien välillä, joka tekee myös erittäin kattavan lokidatan analysoimisesta nopeaa ja suhteellisen helppoa.

7.2 Mahdollisuudet

Tuotteen ensimmäinen mahdollisuus on se, että siihen voi integroida lähes mitä tahansa dataa. Tuote pystyy käsittelemään hyvinkin erilaista dataa ja sillä voidaan käsitellä myös esimerkiksi liiketoimintaan liittyvää dataa. Tuote on todella joustava, mutta datan integrointi

voi vaatia työtä. Toinen mahdollisuus on se, että tuotteessa ei ole lisenssimaksuja, joten kuka tahansa, jolla on tietokone käytettävissään voi hyödyntää tuotetta.

7.3 Uhat

ELk-järjestelmän uhkana on sen hinta. Lisenssi on ilmainen, mutta tuotantokäytössä tarvitaan aina tukisopimus. Tällöin tuote ei ole ilmainen, vaan lisenssin sijasta maksetaan tuesta. Toinen uhka on se, että tuotteen käyttöönotto voi vaatia paljon työtä. Mikäli integroitavat lokilähteet eivät ole suoraan oikeanmuotoisia, niin niiden integroiminen voi vaatia huomattavan määrän työtä, koska dataa varten joudutaan kirjoittamaan käsin täysin manuaalisesti koodia, joka parsii datan oikein näytettävään muotoon. Kolmas uhka on se, että tuotteen integrointeja voidaan tehdä useilla eri tavoilla ja useilla tuotteilla. Tämä on uhka sen takia, että se voi luoda sekavuutta arkkitehtuuriin, sekä vaikeuttaa datan integrointia.

7.4 Heikkoudet

Integraatioiden toteuttaminen on välillä erittäin hankalaa. Suoraan tuetut integraatiot on helppo toteuttaa, mutta mikäli integroidaan muuta dataa se ei ole avain helppoa. Tämä vaatii paljon työtä. Toinen heikkous on se, että hälytyssääntöjen tehokas luonti ja datan korrelointi vaatii maksullisen lisenssin, tai tarkemmin sanoen tukisopimuksen. Ilmaisella versiolla ei voi korreloida dataa, eli tuotetta ei voida käydä tehokkaasti lokilähteiden valvontaan ja hälytyksiin.

Neljäntenä heikkoutena on se, että tuotteen käyttöliittymän kautta ei voi lisätä dataa suoraan. Tämä on muuttumassa ja tulevaisuudessa CSV ja JSON dataa voi lisätä myös käyttöliittymästä suoraan, kuitenkin tällä hetkellä ominaisuuden puute on heikkous, koska se hankaloittaa lokin analysointia manuaalisesti.

8 Johtopäätökset

Tämän opinnäytetyön tarkoitus oli selvittää, voidaanko ELK järjestelmää käyttämällä tehostaa lokien analysoimista ja valvontaa. Opinnäytetyössä käytettävistä menetelmistä erityisen tärkeässä roolissa oli tuotteen asennus ja varsinaisen datan analysoimiseen ja valvontaan kykenevän järjestelmän toteutus.

Opinnäytetyössä pystyttiin käytännön toteutuksella osoittamaan, että ELK järjestelmällä pystytään suhteellisen yksinkertaisesti tehostamaan lokianalyysin tekemistä, sekä lokivalvontaa huomattavan paljon. Järjestelmään voidaan luoda visualisointeja, jotka ovat helpompi ymmärtää kuin rivit tiedostossa. Tuote tarjoaa visualisointien muodossa myös paljon enemmän kuin käyttöliittymän, jolla voidaan tehdä hakuja lokidataan. Visualisoinnit pystyvät näyttämään erittäin suurta määrää dataa ymmärrettävässä muodossa, sekä näyttämään yhdellä vilkaisulla anomalioita suuren datamäärän joukossa.

Tuotteella pystytään luomaan ymmärrettäviä kokonaisuuksia yhdestä lokilähteestä käyttäen useita erilaisia visualisointeja ja tallennettuja hakuja. Näitä kokonaisuuksia hyödyntäen voidaan luoda erittäin monipuolisia koostettuja näkymiä, joita voidaan käyttää todella hyvin lokidatan analysoimiseen ja valvontaan. Näkymien käyttö on erittäin nopeaa ja helppoa, joten lokianalyysi tehostuu huomattavasti, mikäli verrataan useisiin muihin käytössä oleviin järjestelmiin. Tallennetut haut pystyvät näyttämään vain tarpeelliset kentät ja tarjoavat mahdollisuuden analysoida muita kenttiä yhdellä klikkauksella.

ELK-järjestelmää käyttöönotettaessa myös huomattiin, että lokilähteiden integrointi on hyvin erilaista riippuen lokilähteistä. Tiedostopohjaisissa integroinneissa tiedoston muodolla on erityisen paljon merkitystä. Mikäli tiedosto ei ole CSV- tai JSON-formaatissa, niin tiedostojen tuonti järjestelmään saattaa vaatia huomattavan paljon konfigurointia ja sääntöjen luomista käsiteltävää dataa vasten.

Tiedostojen ollessa oletusarvoisessa muodossa datan tuominen on erittäin helppoa. Tämä tulee huomioida, mikäli tuotetta halutaan käyttää yksittäisissä tapauksissa tiedostoissa olevien lokien analysointeihin. Sääntöjen luominen datan käsittelyyn on aikaa vievää, mikäli se ei ole muodossa, jota järjestelmä suoraan ymmärtää. Lisäksi tiedostot tulee viedä järjestelmään aina nojautuen Filebeatiin, eli graafisen käyttöliittymän kautta tuotteeseen ei saada tällä hetkellä tuotua edes CSV tai JSON muotoista dataa. Datan integrointi voikin olla välillä työläämpää, kun varsinaisten näkymien luonti.

Tietoturvavalvonnassa hyödynnetään lähes poikkeuksetta tuotetta, jolla tehdään lokivalvontaa ja korrelointia. Tällaiset ratkaisut tunnetaan lyhenteellä SIEM, joka tulee sanoista Security Information and Event Management. Järjestelmät keräävät vastaavasti lokidataa keskitettyyn paikkaan, kuten tässäkin opinnäytetyössä tehtiin. Opinnäytetyö kuitenkin osoitti, että mikäli ELK järjestelmää halutaan käyttää datan korrelointiin ja hälyttämiseen tehokkaasti, niin tuotteen ilmainen versio ei ole riittävä. Tuotteesta tarvitaan maksullinen tukisopimus, jotta dataa korreloivat ominaisuudet saadaan käyttöön.

Loppujen lopuksi ELK tarjoaa erittäin loistavan mahdollisuuden analysoida lähes mitä vain dataa, kunhan se saadaan integroiduksi sisään järjestelmään. Osa datasta on todella helppo integroida, kuten suorituskykymittaukset, mutta osa vaatii todella paljon erikoisosaamista. Tuotteen paras ominaisuus on visualisoinnit, mutta niiden tekeminen vaatii sen, että tekijä ymmärtää mitä on tekemässä. Visualisoinnit mahdollistavat todella ihmisläheisen tavan tehdä analyysia, mutta niistä voi myös olla enemmän haittaa kuin hyötyä sikäli, jos niitä ei ole tehty ajatuksella.

Tämän opinnäytetyön tutkimustulokset ovat toistettavia vastaavissa olosuhteissa ja lopputuloksissa ei ole suurta vaihtelevuutta. Mikäli toinen tutkija asentaa järjestelmän vastaavaan

ympäristöön käyttäen vastaavia työkaluja, jotka työhön valittiin, niin lopputulos vastaa tämän opinnäytetyön lopputulosta. Tällöin reliabiliteetti täyttyy työn lopputuloksessa.

Opinnäytetyön lopputulos pystyy vastaamaan opinnäytetyössä esitettyihin tutkimuskysymyksiin. Lopputulos pystyy myös osoittamaan, että asennuttu ELK-järjestelmä pystyy valvomaan kotiverkkoa, sekä analysoimaan lokia, joka sisältää satoja miljoonia rivejä dataa. Tällöin lopputulos on toistettava ja tulokset ovat järkeviä, eli myös validiteetti täyttyy työssä.

9 Jatkokehitysehdotukset

Opinnäyteyössä käsiteltiin melko kattavasti muutamia osa-alueita, jotka ovat melko tärkeitä tietoturvavalvonnan näkökulmasta. Aihealue on kuitenkin erittäin laaja, joten koko aiheen käsittely oli käytännössä mahdotonta opinnäytetyössä.

Mikäli tuotetta käytettäisiin tietoturvavalvontaan yritysympäristössä, niin seuraavaksi tulisi testata tuotteen kaupallisen puolen ominaisuuksia. Kaupallisessa versiossa pystytään käyttämään tuotteessa olevia korreloivia mekanismeja, joten datasta voitaisiin luoda automaattisesti myös hälytyksiä perustuen korreloituun dataan. Tämä perustuu tuotteessa olevaan koneoppimiseen.

Tuotteeseen tulisi integroida monipuolisemmin erilaisia lokilähteitä ja opinnäytetyöstä puuttui esimerkiksi pilvipohjaiset järjestelmät kokonaan. Pilvipohjaisia järjestelmiä tulisikin integroida järjestelmään, jotta voitaisiin todeta niiden toimivan yhtä hyvin kuin paikallisesta verkosta tuotettu data. Lisäksi yritysympäristöissä käytetään lähes poikkeuksetta toimialueympäristöä, jonka valvontaa ei tämän opinnäytetyön yhteydessä testattu. Toimialueen valvontaa tulisi testata seuraavaksi, vastaavasti kuin testattiin muitakin lokilähteitä.

ELK:a hyödynnettiin opinnäytetyössä melko monipuolisesti erilaisen datan analysoimiseen ja valvomiseen, mutta näkymät eivät olleet kaikin puolin kovin yksiselitteisiä. Visualisointien tekijä ymmärtää ajatuksen visualisointien takana, mutta mikäli ulkopuolinen henkilö näkee visualisoinnit ilman tarkempia selityksi voi niiden ymmärtäminen olla melko hankalaa. Visualisointeja tulisi kehittää helpommin ymmärrettävään muotoon, sekä loogisempiin kokonaisuuksiin mikäli niitä käytettäisiin tuotantoympäristössä.

Lähteet

Painetut
Alasuutari, P. 2011. Laadullinen tutkimus 2.0. Tampere: Vastapaino

Grönfors M. & Vilkka H. 2011. Laadullisen tutkimuksen kenttätyömenetelmät. Hämeenlinna: SoFia-Sosiologi-Filosofiapu Vilkka 2011.

Kuc, R. & Rogozinski, M. 2014. ElasticSearch Server Second Edition. United Kingdom: Packt Publishing, Limited.

Salonen K. 2013. Näkökulmia tutkimukselliseen ja toiminnalliseen opinnäytetyöhön: opas opiskelijoille, opettajille ja TKI-henkilöstölle. Tampere: Suomen yliopistopaino - Juvenes Print.

Sarjajärvi A. & Tuomi J. 2009. Laadullinen tutkimus ja sisällönanalyysi. Helsinki: Tammi.

Vilkka H. 2015. TUTKI JA KEHITÄ. Jyväskylä: PS-kustannus.

Vuorinen T. 2013. Strategiakirja. Liettua: BALTO print.

Sähköiset
Balaji, 2009. Basics of Forensics Log Analysis. Viitattu 23.11.2019. https://www.paladion.net/blogs/basics-of-forensics-log-analysis

Berman, D. 2019. The complete guide to the ELK stack. Viitattu 23.11.2019. https://logz.io/learn/complete-guide-elk-stack/

Bhargava, R. 2018. Best of 2018: Log Monitoring and Analysis: Comparing ELK, Splunk and Graylog. Viitattu 24.11.2019. https://devops.com/log-monitoring-and-analysis-comparing-elk-splunk-and-graylog/

Crowdstrike. 2019. What is Proactive Threat Hunting? Viitattu 23.11.2019. https://www.crowdstrike.com/epp-101/threat-hunting/

Chand, S. 2019. What is Elastichsearch – Getting Started With No Constraints Search engine. Viitattu 23.11.2019. https://www.edureka.co/blog/what-is-elasticsearch/

Dunham, R. 2019. Logging and Monitoring – An Essential Part of Every Security Program. Viitattu 23.11.2019. https://linfordco.com/blog/logging-and-monitoring/

Elastic. Beats Ligthweight data shippers. Viitattu 23.11.2019. https://www.elastic.co/products/beats

Fitzpatrick, S. Log Monitoring vs Log analysis: What's the difference?. Viitattu 23.11.2019. https://logdna.com/blog/log-monitoring-and-analysis/

GadAllag, S. 2004. The Importance of Logging and Traffic Monitoring for Information Security. Viitattu 23.11.2019. https://www.sans.org/reading-room/whitepapers/logging/paper/1379

Guru99. What is ELK Stack? Viitattu 23.11.2019. https://www.guru99.com/elk-stack-tutorial.html

Haden, P. SOF-ELK. Viitattu 23.11.2019. https://github.com/philhagen/sof-elk

Kroupp, G. Analyzing Log Data: Why it's important? Viitattu 15.10.2019. https://coralogix.com/log-analytics-blog/analyzing-log-data-important/

Kreps, J. 2013. The Log: What every software engineer should know about real-time data's unifying abstraction. Viitattu 23.11.2019. https://engineering.linkedin.com/distributed-systems/log-what-every-software-engineer-should-know-about-real-time-datas-unifying

Logdna. What is Log Management? The Complete Logging Guide. Viitattu 23.11.2019. https://logdna.com/what-is-log-management/

Logz.io. 2019. Logz.io pricing. Viitattu 24.11.2019. https://logz.io/pricing/

McRee, R. 2017. Adversary hunting with SOF-ELK. Viitattu 24.11.2019. https://holisticinfosec.blogspot.com/2017/07/toolsmith-126-adversary-hunting-with.html

Miller, J. 2019. What is Security Logging and Monitoring? Viitattu 23.11.2019. https://www.bitlyft.com/what-is-security-logging-and-monitoring/

Moloch. Tuotteen kotisivu. Viitattu 11.12.2019. https://molo.ch/

Objectrocket. What are Elasticsearch Beats? Viitattu 23.11.2019. https://www.objectrocket.com/resource/what-are-elasticsearch-beats/

Opintokeskus Sivis. SWOT-analyysi. Viitattu 10.12.2019. https://www.ok-sivis.fi/jarjestoarvioinnin-ilmansuuntia/arvioinnin-tiedonkeruun-menetelmia/swot-analyysi.html

Rovnik, V. Reporting and data visualization in Kibana. Viitattu 20.12.2019. https://towardsdatascience.com/reporting-data-visualization-in-kibana-336bbe92ac9d

Security Onion. Security Onion Documentation. Viitattu 24.11.2019. https://security-onion.readthedocs.io/en/latest/index.html

Sumo Logic. Log Analysis. Viitattu 23.11.2019. https://www.sumologic.com/glossary/log-analysis/

Ubiquiti. EdgeRouter – Remote Syslog Server for System Logs. Viitattu 3.12.2019. https://help.ubnt.com/hc/en-us/articles/204975904-EdgeRouter-Remote-Syslog-Server-for-System-Logs

Wazuh. Web-sivu. Viitattu 15.12.2019. https://wazuh.com/

Zhang, E. 2018.What is Log Analysis? Use Cases, Best Practices, and more. Viitattu 23.11.2019. https://digitalguardian.com/blog/what-log-analysis-use-cases-best-practices-and-more

Kuviot

Kuvio 1: Lokitapahtuman kirjoitus lokiin järjestyksessä (Kreps 2013.) 9

Kuvio 2: ELK järjestelmän komponentit. (Berman 2019.) 17

Kuvio 3: SOF-ELK httpd lokin visualisointinäkymä (Mcree). 22

Kuvio 4: Security Onionin Kibana näkymä. Kuva omasta ympäristöstä. 24

Kuvio 5: Valvottavan verkon arkkitehtuurikuva. 27

Kuvio 6: Yksinkertaistettu ELK arkkitehtuuri. 28

Kuvio 7: Käytettävä arkkitehtuuri. (Elastic.) 29

Kuvio 8: Elasticsearch datan siivousasetukset. 31

Kuvio 9: Siivouspolitiikan käyttöönotto kaikkiin indekseihin. 32

Kuvio 10: Logstash yleinen konfiguraatio. 33

Kuvio 11: Ubiquiti syslog kriittisyysasetikko. (Ubiquiti.) 34

Kuvio 12: Filebeat konfiguraatio. 34

Kuvio 13: Logstash konfiguraatio reitittimen Syslogille. 35

Kuvio 14: Auditbeat asetukset. 37

Kuvio 15: Metricbeatin konfiguraatio. 38

Kuvio 16: Kibana hakujen oletusnäkymä. 39

Kuvio 17: Sisäänkirjautumisnäkymä. 39

Kuvio 18: Tallennettu haku DNS kyselyjen tarkastamiseen. 40

Kuvio 19: Päätelaitevalvonnan tapahtumien määrä. 41

Kuvio 20: Verkkosensorin hälytystiedoista luotu visualisointi. 41

Kuvio 21: Verkkosensorin hälytysaikajana. 42

Kuvio 22: Kotinäkymä rakennetusta ympäristöstä. 43

Kuva 23: Opinnäyteyötä varten luotu connections dashboard. 44

Kuvio 24: Connections dashboardin tapahtumien määrän visualisointi. 44

Kuvio 25: Yleisimmät kohdeportit visualisointi. 45

Kuvio 26: Yleisimpien kohdepalvelujen visualisointi. 45

Kuvio 27: Visualisointi, joka näyttää maat joihin yhteyksiä on muodostettu eniten..... 46

Kuvio 28: Visualisointi, joka kuvaa maita joihin yhteyksiä on vähiten. 46

Kuvio 29: Visualisointi verkkotapahtumien aikajanasta jaoteltuna palvelulla. 47

Kuvio 30: Tallennettu haku, joka kuvaa verkkoyhteyksiä. 47

Kuvio 31: Datan suodatus visualisoinnin sisältä. 47

Kuvio 32: Lokilähteiden arkkitehtuuri. 49

Kuvio 33: SWOT-analyysi..... 50

Taulukot

Taulukko 1: ELK Beatsien käyttötarkoitus. (Elasticsearch.) 17

Taulukko 2: ELK-järjestelmien opinnäytetyön kannalta tärkeät ominaisuudet. 25

Taulukko 3: ELK asennuksen virtuaalipalvelinten resurssit rooleittain. 29

Taulukko 4: Lokilähteiden integroinnit. 48